Premium et Atrium sous Unity Pro

Bus de terrain CANopen
Manuel utilisateur

07/2012

35008148.10

www.schneider-electric.com

Le présent document comprend des descriptions générales et/ou des caractéristiques techniques des produits mentionnés. Il ne peut pas être utilisé pour définir ou déterminer l'adéquation ou la fiabilité de ces produits pour des applications utilisateur spécifiques. Il incombe à chaque utilisateur ou intégrateur de réaliser l'analyse de risques complète et appropriée, l'évaluation et le test des produits pour ce qui est de l'application à utiliser et de l'exécution de cette application. Ni la société Schneider Electric ni aucune de ses sociétés affiliées ou filiales ne peuvent être tenues pour responsables de la mauvaise utilisation des informations contenues dans le présent document. Si vous avez des suggestions d'amélioration ou de correction ou avez relevé des erreurs dans cette publication, veuillez nous en informer.

Aucune partie de ce document ne peut être reproduite sous quelque forme ou par quelque moyen que ce soit, électronique, mécanique ou photocopie, sans l'autorisation écrite expresse de Schneider Electric.

Toutes les réglementations locales, régionales et nationales pertinentes doivent être respectées lors de l'installation et de l'utilisation de ce produit. Pour des raisons de sécurité et afin de garantir la conformité aux données système documentées, seul le fabricant est habilité à effectuer des réparations sur les composants.

Lorsque des équipements sont utilisés pour des applications présentant des exigences techniques de sécurité, suivez les instructions appropriées.

La non-utilisation du logiciel Schneider Electric ou d'un logiciel approuvé avec nos produits matériels peut entraîner des blessures, des dommages ou un fonctionnement incorrect.

Le non-respect de cette consigne peut entraîner des lésions corporelles ou des dommages matériels.

© 2012 Schneider Electric. Tous droits réservés.

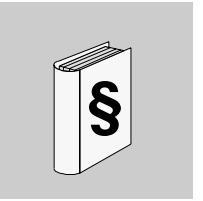
Table des matières

	Consignes de sécurité	**7**
	A propos de ce manuel	**9**
Chapitre 1	**Général**	**11**
	Principes	12
	Architecture générale d'un bus de terrain CANopen	13
	Vitesse de transmission et longueur de câble	15
	Limitations relatives aux câbles de dérivation	17
	Vue d'ensemble des phases d'installation	19
Chapitre 2	**Présentation de la carte PCMCIA TSX CPP 110**	**21**
2.1	Description de la carte TSX CPP 110	22
	Informations relatives à la carte TSX CPP 110	23
	Fixation de la carte TSX CPP 110	25
2.2	Spécifications techniques	27
	Normes et caractéristiques	28
	Caractéristiques CANopen	29
	Processeurs prenant en charge la carte TSX CPP 110	30
Chapitre 3	**Configuration logiciel**	**31**
3.1	Généralités	32
	Principe de mise en oeuvre	33
	Méthode de mise en oeuvre	34
	Adressage topologique CANopen	35
3.2	Configuration	36
	Accès à l'écran de configuration	37
	Ecran de configuration de la carte TSX CPP 110 PCMCIA	39
	Description des données d'E/S et du comportement du bus au démarrage	42
	Procédure de sélection du mode de chargement de la configuration	45
	Comment effectuer un chargement de configuration par X-Way	47
	Ecran de configuration des esclaves	49
	Procédure de configuration de la carte PCMCIA CANopen	50
	Bus CANopen affiché dans le navigateur de projet	53
	Fichier de configuration de la carte TSX CPP 110 PCMCIA	54

3.3 Programmation 56
Accès aux SDO CANopen 57
Demande IDENTIFICATION 63
Commandes diagnostics 65
3.4 Mise au point 67
Description de l'écran de mise au point 67
3.5 Diagnostics 69
Diagnostics à l'aide des voyants d'état sur la carte PCMCIA TSX CPP 110 70
Données de diagnostic 71
Comment effectuer un diagnostic 74
Chapitre 4 Objets langage CANopen 77
4.1 Objets langage et IODDT pour la communication CANopen 78
Présentation des objets langage pour la communication CANopen 79
Objets langage à échange implicite associés à la fonction métier 80
Objets langage à échange explicite associés à la fonction métier 81
Gestion des échanges et comptes rendus avec des objets explicites . . . 83
4.2 Objets langage et IODDT générique applicables aux protocoles de communication 87
Détails des objets à échange implicite de type d'IODDT T_COM_STS_GEN 88
Détails des objets à échange explicite de l'IODDT de type T_COM_STS_GEN 89
4.3 Objets langage du IODDT spécifique à CANopen 91
Détails des objets à échange implicite de type T_COM_CPP110 de l'IODDT 92
Détails des objets à échange implicite de type T_COM_CPP110 n'appartenant pas à l'IODDT 96
Objets langage à échange explicite de l'IODDT T_COM_CPP110 97
4.4 IODDT de type T_GEN_MOD applicable à tous les modules 98
Présentation des objets langage de l'IODDT de type T_GEN_MOD 98
4.5 Objets langage de configuration CANopen 100
Objets langage associés à la configuration 100
4.6 Codes d'erreur CANopen 102
Codes d'erreur CANopen 102
Chapitre 5 Exemples d'installation de bus CANopen 109
5.1 Description de l'exemple 110
Description de l'exemple 110
5.2 Mise en oeuvre matérielle 111
Configuration matérielle îlots Advantys 112
Configuration matérielle du maître 115
Configuration matérielle du bus 116

5.3 Mise en oeuvre logicielle 119
Configuration logicielle Advantys 120
Déclaration du maître CANopen sous Sycon et import EDS 126
Configuration du bus CANopen. 129
Déclaration des noeuds 2 et 3. 130
Configuration des noeuds 2 et 3 132
Configuration de la carte PCMCIA TSX CPP 110. 134
Mise au point 139
Annexes 141
Annexe A Exemple de configuration pour équipements sur le bus CANopen 143
Configuration d'un variateur de vitesse Altivar 144
Configuration d'un variateur Lexium 147
Configuration de plus de 4 PDO par nœud 150
Glossaire 153
Index 157

Consignes de sécurité



Informations importantes

AVIS

Lisez attentivement ces instructions et examinez le matériel pour vous familiariser avec l'appareil avant de tenter de l'installer, de le faire fonctionner ou d'assurer sa maintenance. Les messages spéciaux suivants que vous trouverez dans cette documentation ou sur l'appareil ont pour but de vous mettre en garde contre des risques potentiels ou d'attirer votre attention sur des informations qui clarifient ou simplifient une procédure.

La présence d'un de ces symboles sur une étiquette de sécurité Danger collée sur un équipement indique qu'un risque d'électrocution existe, susceptible d'entraîner la mort ou des blessures corporelles si les instructions ne sont pas respectées.

Ce symbole est le symbole d'alerte de sécurité. Il vous avertit d'un risque de blessures corporelles. Respectez scrupuleusement les consignes de sécurité associées à ce symbole pour éviter de vous blesser ou de mettre votre vie en danger.

⚠ DANGER

DANGER indique une situation immédiatement dangereuse qui, si elle n'est pas évitée, **entraînera** la mort ou des blessures graves.

⚠ AVERTISSEMENT

AVERTISSEMENT indique une situation potentiellement dangereuse et **susceptible d'entraîner** la mort ou des blessures graves.

⚠ ATTENTION

ATTENTION indique une situation potentiellement dangereuse et **susceptible d'entraîner** des blessures mineures ou modérées.

AVIS

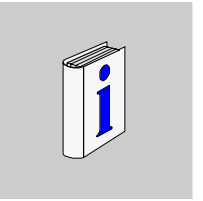
AVIS indique des pratiques n'entraînant pas de risques corporels.

REMARQUE IMPORTANTE

L'installation, l'utilisation, la réparation et la maintenance des équipements électriques doivent être assurées par du personnel qualifié uniquement. Schneider Electric décline toute responsabilité quant aux conséquences de l'utilisation de ce matériel.

Une personne qualifiée est une personne disposant de compétences et de connaissances dans le domaine de la construction, du fonctionnement et de l'installation des équipements électriques, et ayant suivi une formation en sécurité leur permettant d'identifier et d'éviter les risques encourus.

A propos de ce manuel



Présentation

Objectif du document

Ce manuel présente la communication CANopen sur les automates Premium et Atrium.

Champ d'application

Ce document est applicable à partir de Unity Pro 7.0.

Information spécifique au produit

⚠ AVERTISSEMENT

COMPORTEMENT INATTENDU DE L'EQUIPEMENT

L'utilisation de ce produit requiert une expertise dans la conception et la programmation des systèmes de contrôle. Seules les personnes avec l'expertise adéquate sont autorisées à programmer, installer, modifier et utiliser ce produit.

Respectez toutes les réglementations et normes de sécurité locales et nationales.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Commentaires utilisateur

Envoyez vos commentaires à l'adresse e-mail techpub@schneider-electric.com

Général

1

Objet de ce chapitre

Ce chapitre présente les principales caractéristiques techniques d'une communication CANopen.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Principes	12
Architecture générale d'un bus de terrain CANopen	13
Vitesse de transmission et longueur de câble	15
Limitations relatives aux câbles de dérivation	17
Vue d'ensemble des phases d'installation	19

Principes

Introduction

Développé à l'origine pour les systèmes embarqués des véhicules automobiles, le bus de communication CAN est maintenant utilisé dans de nombreux domaines comme :

- le transport ;
- les équipements mobiles ;
- les équipements médicaux ;
- le bâtiment ;
- le contrôle industriel.

Les points forts du système CAN sont :

- son système d'allocation du bus ;
- la détection des erreurs ;
- la fiabilité des échanges de données.

Structure maître/esclave

Le bus CAN possède une structure maître/esclave pour la gestion du bus.

Le maître gère :

- l'initialisation des esclaves ;
- les erreurs de communication ;
- le statut des esclaves.

Communication poste à poste

La communication se fait poste à poste, ce qui permet à chaque équipement d'envoyer à tout moment une demande au bus à laquelle les équipements concernés répondent. La priorité des requêtes circulant sur le bus est déterminée par un identifiant pour chaque message.

Identificateurs CAN :

Les échanges explicites au niveau des liaisons utilisent des identificateurs au format étendu (29 bits) (CAN V2.0B standard).

Les identificateurs 11 bits (CAN V2.0A standard) peuvent également être utilisés pour l'envoi de demandes mais ce type d'identificateurs ne peut être reçu.

Architecture générale d'un bus de terrain CANopen

Présentation

L'architecture ouverte CANest composée des éléments suivants :

- un maître du bus (carte PCMCIA TSX CPP 110),
- des équipements esclaves.

NOTE : Il est également possible de connecter plusieurs cartes PCMCIA TSX CPP 110 sur le bus, une en tant que maître et toutes les autres en mode **écoute**. Les cartes en mode **écoute** permettent aux automates Premium auxquels elles sont connectées de connaître le statut du bus et des esclaves du bus à tout moment.

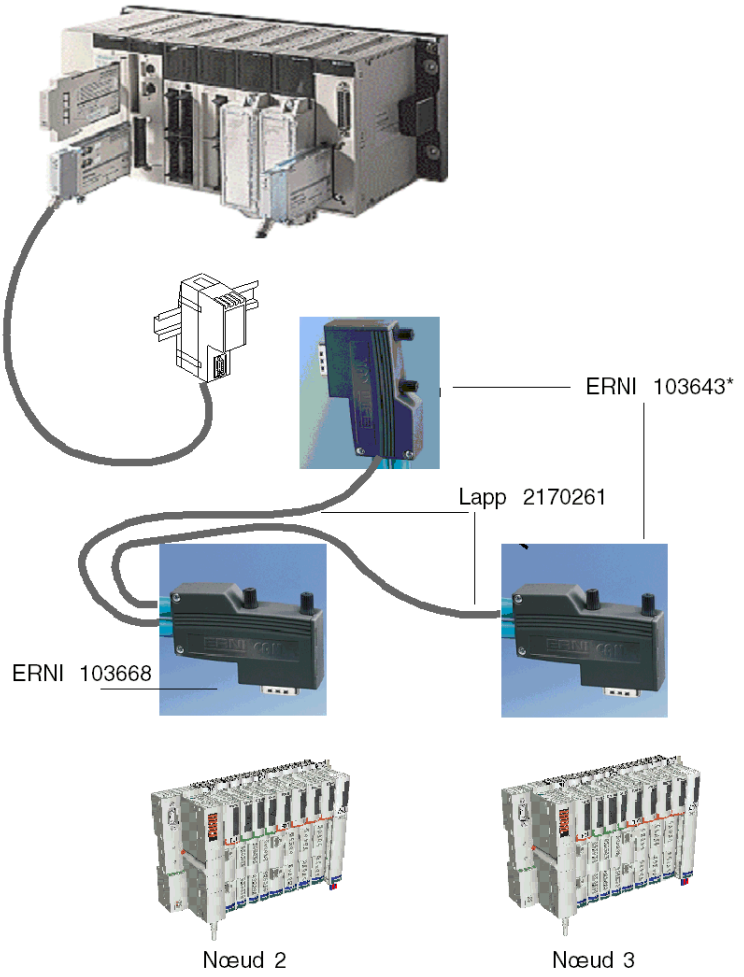
CANopen accepte 128 équipements (c'est-à-dire le maître du bus et 127 esclaves distants).

La vitesse de transmission dépend strictement de la longueur du bus et du type de câble utilisé.

NOTE : Pour configurer un bus CANopen, utilisez le logiciel Sycon version 2.8 ou supérieure.

Exemple de bus

La figure ci-dessous montre un exemple d'un bus CANopen qui sera décrit en détails à la fin de ce document dans l'exemple de mise en œuvre d'un bus CANopen.



*: impédance de ligne de terminaison intégrée (120 Ohms).

Pour plus de détails, reportez-vous au document "Manuel d'installation de matériel CANopen" sur telemecanique.com

Vitesse de transmission et longueur de câble

Vue d'ensemble

CANopen prend en charge 127 dispositifs (le bus maître et 126 esclaves distants). La vitesse de transmission dépend strictement du type de câble utilisé.

Dans le protocole CAN, la priorité des trames est gérée par une collision entre des niveaux dominants et récessifs de la ligne. Cette collision doit être résolue durant la transmission d'un bit, ce qui limite le délai de propagation du signal entre deux nœuds.

Les tableaux suivants indiquent la longueur maximale du câble principal en fonction du câble CANopen fourni par Schneider Electric (TSXCANCA•••, TSXCANCB•••, TSXCANCD•••).

Longueur maximale du câble

En conséquence, la distance maximale entre les deux nœuds les plus éloignés d'un bus CAN dépend de la vitesse et est fournie dans le tableau suivant :

Vitesse en bits/s	Longueur maximale du câble
1 Mbits/s	20 m (65 ft)
800 Kbits/s	40 m (131 ft)
500 Kbits/s	100 m (328 ft)
250 Kbits/s	250 m (820 ft)
125 Kbits/s	500 m (1 640 ft)
50 Kbits/s	1 000 m (3 280 ft)
20 Kbits/s	2 500 m (8 202 ft)
10 Kbits/s	5 000 m (16 404 ft)

Conformément à la stratégie réseau Schneider Electric, les vitesses 1 Mbits/s, 800 Kbits/s, 500 Kbits/s, 250 Kbits/s et 125 Kbits/s sont recommandées pour les solutions d'automatisation au niveau machine et installation.

NOTE : la longueur maximale suppose un temps de propagation interne à l'équipement raisonnable et un point d'échantillonnage binaire. Des temps de propagation interne importants limitent en effet la longueur de câble maximale qu'il est possible d'obtenir pour l'équipement.

Les longueurs de câbles du tableau ci-dessus peuvent inclure un câble de dérivation à l'extrémité physique du câble principal.

Répéteurs diminuant la longueur de câble

Les valeurs ci-dessus définissent la longueur maximale de câble sans répéteur. Les répéteurs ajoutent un temps de propagation dans le bus, ce qui entraîne une réduction de la longueur maximale de ce dernier. Un temps de propagation de 5 ns entraîne une réduction de longueur de 1 m (3 ft).

Exemple : un répéteur avec un délai de propagation de 150 ns diminue la longueur maximale de câble de 30 m (98 ft).

Longueur maximale de câble et nombre de nœuds

Outre les limitations de longueur liées à la vitesse de transmission, la longueur maximale de câble dépend également de la résistance de charge.

Dans tous les cas, le nombre maximal de nœuds pouvant être connectés au même segment est limité à 64. Pour connecter davantage de nœuds à un segment, utilisez un répéteur.

Le tableau ci-après explique l'incidence du nombre de nœuds sur la longueur de câble.

Nombre de nœuds	Longueur maximale du câble
2	229 m (751,31 ft)
16	210 m (688,97 ft)
32	195 m (639,76 ft)
64	170 m (557,74 ft)

Isolation des équipements CANopen

Dans les documents traitant de CANopen, vous trouverez souvent une valeur maximale de 40 m (131 ft) pour une vitesse de transmission de 1 Mbits/s. Cette longueur est calculée sans isolation telle que prévue dans les équipements CANopen Schneider Electric.

Avec une telle isolation, la longueur minimale de réseau calculée est de 4 m (13 ft) pour une vitesse de transmission de 1 Mbits/s. Toutefois, l'expérience donne en pratique une longueur de 20 m (65 ft) qui peut être éventuellement diminuée par des dérivations ou d'autres influences.

Limitations relatives aux câbles de dérivation

Présentation

Un câble de dérivation crée une réflexion du signal sur la caractéristique de ligne de transmission du câble principal. Pour minimiser les réflexions, limitez autant que possible la longueur des câbles de dérivation.

Longueur maximale de câble de dérivation

Respectez les valeurs du tableau ci-après.

Vitesse de transmission	Lmax	ΣLmax	Distance boîtier	ΣLGmax
1 Mbits/s	0,3 m (0,98 ft)	0,6 m (1,96 ft)		1,5 m (4,92 ft)
800 Kbits/s	3 m (9,84 ft)	6 m (19,68 ft)	3,6 m (11,81 ft)	15 m (49,21 ft)
500 Kbits/s	5 m (16,4 ft)	10 m (32,8 ft)	6 m (19,68 ft)	30 m (98,42 ft)
250 Kbits/s	5 m (16,4 ft)	10 m (32,8 ft)	6 m (19,68 ft)	60 m (196,84 ft)
125 Kbits/s	5 m (16,4 ft)	10 m (32,8 ft)	6 m (19,68 ft)	120 m (393,69 ft)
50 Kbits/s	60 m (196,84 ft)	120 m (393,69 ft)	72 m (236,21 ft)	300 m (984,24 ft)
20 Kbits/s	150 m (492,12 ft)	300 m (984,24 ft)	180 m (590,54 ft)	750 m (2 460,62 ft)
10 Kbits/s	300 m (984,24 ft)	600 m (1 968,49 ft)	360 m (1 181,09 ft)	1 500 m (4 921,24 ft)

Lmax est la longueur maximale d'un câble de dérivation.

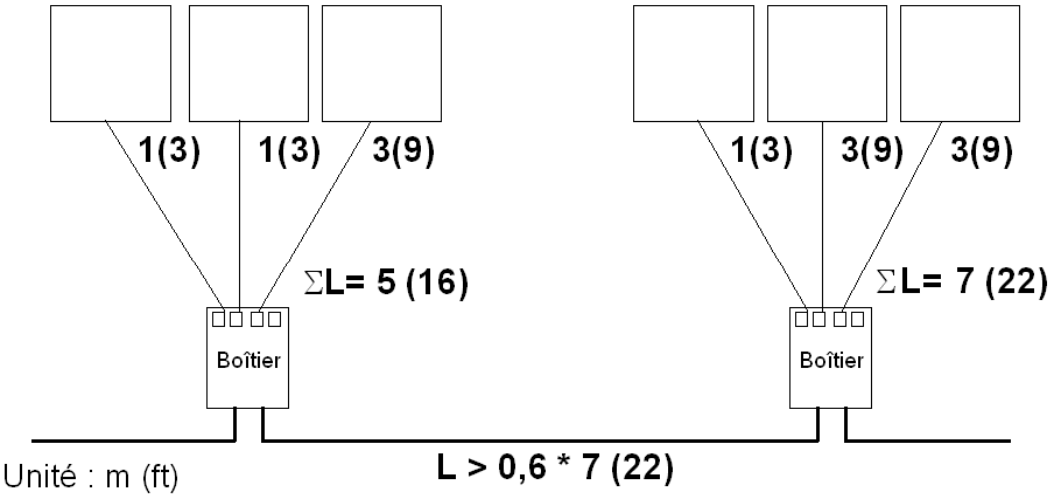
Σ**Lmax** est la valeur maximale de la somme de tous les câbles de dérivation sur le même boîtier de dérivation (TAP).

Distance boîtier est la distance minimale nécessaire entre deux boîtiers de dérivation. Elle peut être calculée pour chaque boîtier (doit être supérieure à 60 % de la plus grande des deux valeurs ΣLmax).

Σ**LGmax** est la valeur maximale de la somme de tous les câbles de dérivation sur le réseau.

Exemple de calcul

Le schéma ci-après fournit un exemple de calcul de distance de boîtier avec 2 boîtiers de dérivation et 6 équipements :



La distance de boîtier dans l'exemple ci-dessus est calculée comme suit :

Etape	Description	Résultat
1	Calcul de la somme des longueurs des câbles de dérivation pour chaque boîtier de dérivation.	5 m (16 ft) et 7 m (22 ft)
2	Sélection de la plus grande longueur.	7 m (22 ft)
3	Calcul de la longueur de câble minimale entre les deux boîtiers de dérivation.	60 % de 7 m (22 ft)

Respectez la distance boîtier, même si un équipement est intercalé.

Vue d'ensemble des phases d'installation

Introduction

La mise en œuvre logicielle des modules métier est réalisée depuis les différents éditeurs de Unity Pro :

- en mode local ;
- en mode connecté.

Si vous ne disposez pas de processeur auquel vous pouvez vous connecter, Unity Pro vous permet d'effectuer un test initial à l'aide du simulateur. Dans ce cas, l'installation *(voir page 20)* est différente.

Il est recommandé de respecter l'ordre des phases d'installation suivant. Toutefois, il est possible de modifier cet ordre (en commençant par la phase de configuration, par exemple).

Phases d'installation à l'aide du processeur

Le tableau ci-dessous présente les différentes phases d'installation avec le processeur :

Phase	Description	Mode
Déclaration des variables	Déclaration des variables de type IODDT pour les modules métier et les variables du projet.	Local (1)
Programmation	Programmation du projet.	Local (1)
Configuration	Déclaration des modules.	Local
	Configuration des voies du module.	
	Saisie des paramètres de configuration.	
Association	Association des variables IODDT aux voies configurées (éditeur de variables).	Local (1)
Génération	Génération du projet (analyse et édition des liens).	Local
Transfert	Transfert du projet vers l'automate.	Connecté
Réglage/Mise au point	Mise au point du projet depuis les écrans de mise au point et les tables d'animation.	Connecté
	Modification du programme et des paramètres de réglage.	
Documentation	Création d'un fichier de documentation et impression des diverses informations relatives au projet.	Connecté (1)
Fonctionnement/ Diagnostic	Affichage des diverses informations nécessaires à la supervision du projet.	Connecté
	Diagnostic du projet et des modules.	
Légende :		
(1)	Ces différentes phases peuvent aussi être réalisées dans l'autre mode.	

Phases de mise en œuvre à l'aide du simulateur

Le tableau ci-dessous présente les différentes phases de mise en œuvre avec le simulateur :

Phase	Description	Mode
Déclaration des variables	Déclaration des variables de type IODDT pour les modules métier et les variables du projet.	Local (1)
Programmation	Programmation du projet.	Local (1)
Configuration	Déclaration des modules.	Local
	Configuration des voies du module.	
	Saisie des paramètres de configuration.	
Association	Association des variables IODDT aux modules configurés (éditeur de variables).	Local (1)
Génération	Génération du projet (analyse et édition des liens).	Local
Transfert	Transfert du projet vers le simulateur.	Connecté
Simulation	Simulation du programme sans les entrées/sorties.	Connecté
Réglage/Mise au point	Mise au point du projet depuis les écrans de mise au point et les tables d'animation.	Connecté
	Modification du programme et des paramètres de réglage.	
Légende :		
(1)	Ces différentes phases peuvent aussi être réalisées dans l'autre mode.	

NOTE : Le simulateur prend uniquement en charge les modules TOR ou analogiques.

Présentation de la carte PCMCIA TSX CPP 110

2

Objet de ce chapitre

Ce chapitre présente les principales caractéristiques techniques d'une carte PCMCIA TSX CPP 110.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
2.1	Description de la carte TSX CPP 110	22
2.2	Spécifications techniques	27

2.1 Description de la carte TSX CPP 110

Objet de ce sous-chapitre

Ce sous-chapitre décrit l'aspect physique de la carte PCMCIA TSX CPP 110 et sa connectique.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Informations relatives à la carte TSX CPP 110	23
Fixation de la carte TSX CPP 110	25

Informations relatives à la carte TSX CPP 110

Présentation

La carte de communication CANopen TSX CPP 110 est utilisée pour mettre en œuvre une architecture ouverte CAN. Cette carte est le maître du bus et permet la connexion d'équipements conformément à la norme CANopen :

- Echange implicite d' **O**bjets de **D**onnées **P**rocess utilisant des mots `%MW`.
- Echange explicite d'**O**bjets de **D**onnées **S**ervice via les blocs fonction `READ_VAR` et `WRITE_VAR`.
- Compatibilité avec les profils d'équipement normalisés et la communication sur CANopen (2.0A et 2.0B).

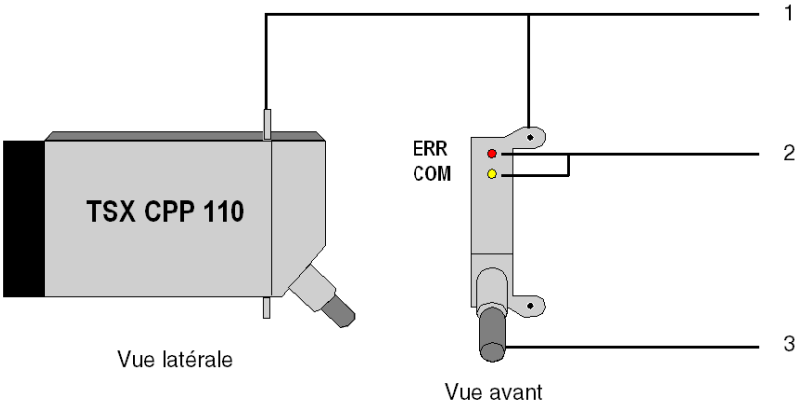
NOTE : La carte TSX CPP 110 permet une configuration maximale de 64 KB.

Description physique

La carte TSX CPP 110 est une carte PCMCIA de type III qui est insérée dans le slot communication PCMCIA sur le processeur.

Les cartes CANopen PCMCIA Type III (TSX CPP 110) peuvent être utilisées dans les emplacements PCMCIA de tous les automates programmables via le logiciel Unity Pro.

Ce module est composé des éléments suivants :



Description

Ce tableau décrit les éléments de la figure précédente.

Nombre	Description
1	Des supports de fixation, en haut et en bas de la carte, qui servent à fixer la carte au processeur.
2	Des voyants, qui montrent les opérations de la carte de communication *(voir page 70)*.
3	Un câble bus, ce câble de 50 cm est terminé par une prise industrielle CANopen.

Fixation de la carte TSX CPP 110

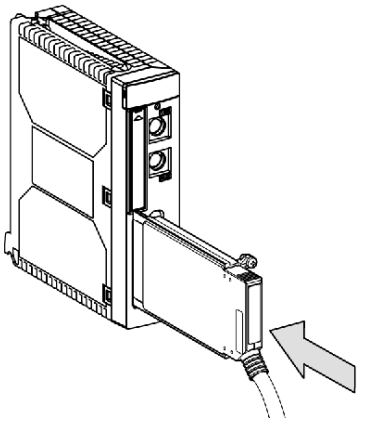
Carte PCMCIA TSX CPP 110

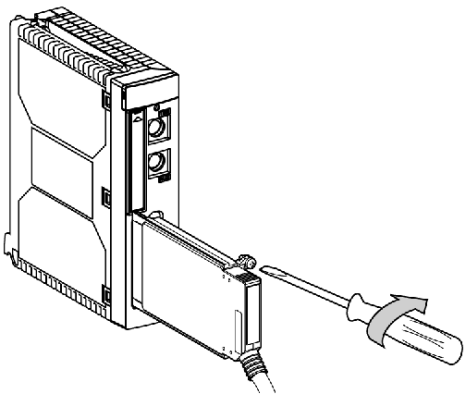
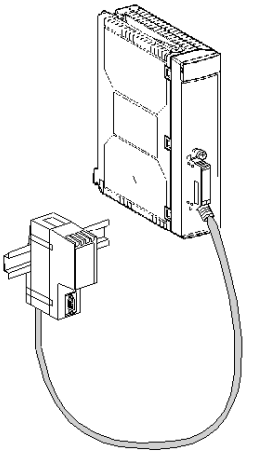
La carte PCMCIA TSX CPP 110 équipée de prise industrielle sert de liaison entre l'UC Premium et le réseau CANopen.

NOTE : le module de communication Modbus TSX SCY 2160• ne peut pas être utilisé.

⚠ ATTENTION
ENDOMAMGEMENT DE LA CARTE PCMCIA Pour insérer ou retirer une carte PCMCIA, l'automate doit être hors tension. **Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.**

Le tableau suivant décrit la procédure d'installation d'une carte TSX CPP 110.

Etape	Opération	Illustration
1	Mettez l'automate hors tension.	
2	Insérez la carte PCMCIA type III dans l'emplacement PCMCIA de l'UC Premium.	

Etape	Opération	Illustration
3	Vissez la carte correctement pour un fonctionnement correct.	
4	Fixez la prise sur un rail DIN.	
5	Remettez l'automate sous tension.	

2.2 Spécifications techniques

Objet de ce sous-chapitre

Ce sous-chapitre décrit les principales caractéristiques techniques de la carte TSX CPP 110.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Normes et caractéristiques	28
Caractéristiques CANopen	29
Processeurs prenant en charge la carte TSX CPP 110	30

Normes et caractéristiques

Normes

La carte de communication TSX CPP 110 est conforme aux normes internationales suivantes :

Normes internationales	ISO IS 11898, Emetteur-récepteur grande vitesse CAN et couche de liaison de données
Normes américaines	UL508
Normes radiations	Label CE, FCC-B (50082-1)

Caractéristiques électriques

- Alimentation V c.a. logique : 5 V fournis par l'alimentation rack
- Consommation : 3 W

Caractéristiques environnementales

- Température de stockage : -25 °C à + 70 °C
- Température de fonctionnement : -0 °C à + 70 °C
- Hydrométrie de stockage : 30 % à 95 % sans condensation
- Hydrométrie de fonctionnement : 5 % à 95 % sans condensation

Caractéristiques CANopen

Normes

La carte de communication TSX CPP 110 est conforme à la norme DS301 V4.01.

Fonctions spécifiques

- L'utilisateur peut utiliser une affectation de contenu PDO conforme à la norme DS301 V4.01.
- La carte TSX CPP 110 prend en charge la fonction "heartbeat" (DS 301V4.01).
- La carte TSX CPP 110 est normalement le maître de la gestion réseau (NMT_MASTER) sur le bus (cette fonction peut être désactivée via SyCon).
- La carte TSX CPP 110 produit normalement la variable de synchronisation (SYNC) (cette fonction peut être désactivée via SyCon).
- L'ID nœud de la carte TSX CPP 110 ne peut pas être utilisé pour le transfert de données. Il est uniquement utilisé pour la fonction "Heartbeat".

Processeurs prenant en charge la carte TSX CPP 110

Présentation

Tous les processeurs Premium et Atrium prennent en charge la carte PCMCIA CANopen.

La carte est mise en œuvre à l'aide du logiciel Unity Pro.

La configuration générale d'un bus CANopen est définie dans le logiciel Sycon 2.8 ou version ultérieure (TLX LFBCM).

NOTE : la carte PCMCIA doit impérativement être installée dans l'emplacement prévu à cet effet dans le module processeur. Un seul bus CANopen est donc disponible pour chaque UC d'automate.

Types de processeurs et capacités

Le tableau ci-dessous détaille les processeurs prenant en charge la carte PCMCIA CANopen TSX CPP 110, ainsi que leurs capacités de stockage maximales.

Processeur	**Taille maximale des données de configuration TSX CPP 110 situées dans le processeur(1)**	**Taille maximale des données d'E/S pour la configuration des nœuds CANopen**	
		Tâche MAST	**Tâche FAST**
TSX P57 0•• TSX P57 1•• TSX P57 1•••	12 Ko	384 %MW (192+192)	48 %MW (24+24)
TSX P57 2•• TSX P57 2••• TSX PCI 57 204	16 Ko	512 %MW (256+256)	64 %MW (32+32)
TSX P57 3•• TSX P57 3••• TSX PCI 57 354	32 Ko	1024 %MW (512+512)	128 %MW (64+64)
TSX P57 4•• TSX P57 4••• TSX P57 5•• TSX P57 5••• TSX P57 6•••	64 Ko	3584 %MW (1792+1792)	512 %MW (2) (256+256)
Notes	(1) : cette taille maximale peut être dépassée si vous chargez la configuration dans la carte avec le logiciel Sycon *(voir page 45)*. La taille maximale des données de configuration autorisée par le logiciel Sycon est de 64 Ko. (2) : la taille maximale des données d'E/S pour la configuration des nœuds CANopen (tâche FAST) est de 1024 %MW (512+512) pour les modules TSX P57 554, TSX P57 5634 et TSX P57 6634 avec le micrologiciel V3.0.		

NOTE : le taux de saturation de la configuration est indiqué dans le mot %KWm.1.2 *(voir page 100)*.

Configuration logiciel

3

Objet de ce chapitre

Ce chapitre décrit les différentes possibilités de configuration, de supervision et de diagnostic d'une application CANopen.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
3.1	Généralités	32
3.2	Configuration	36
3.3	Programmation	56
3.4	Mise au point	67
3.5	Diagnostics	69

3.1 Généralités

Objet de ce sous-chapitre

Ce sous-chapitre décrit la mise en oeuvre logicielle d'une carte PCMCIA TSX CPP 110.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Principe de mise en oeuvre	33
Méthode de mise en oeuvre	34
Adressage topologique CANopen	35

Principe de mise en oeuvre

Présentation

La mise en œuvre d'un bus CAN nécessite de définir le contexte physique de l'application dans laquelle le bus est intégré (rack, alimentation, processeur, modules ou équipements, etc.), puis d'en assurer la mise en œuvre logicielle.

Cette mise en œuvre logicielle est réalisée depuis les différents éditeurs Unity Pro :

- Soit en mode local,
- Soit en mode connecté : dans ce cas, les modifications se limitent à certains paramètres.

Le bus est configuré via le logiciel Sycon.

Principe de mise en œuvre

Le tableau ci-dessous présente les différentes phases de mise en œuvre.

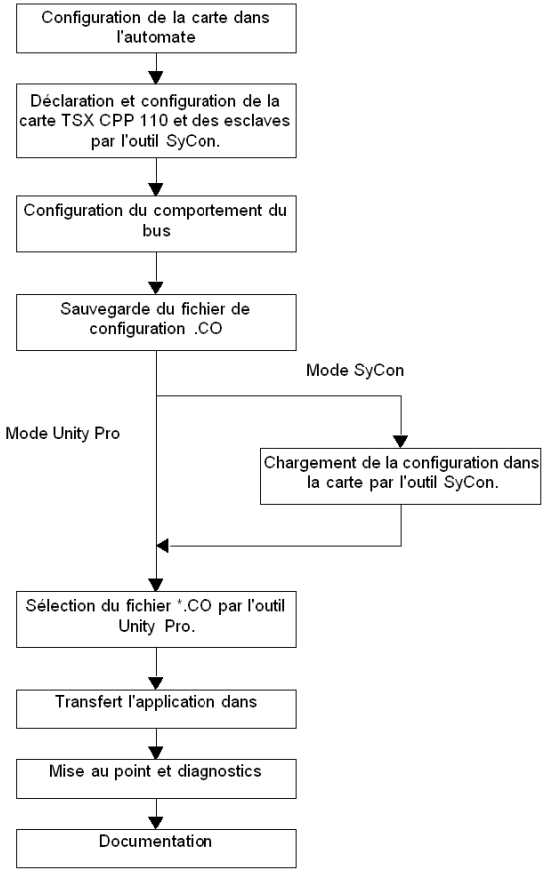
Mode	Phase	Description
Hors ligne	Déclaration de la carte TSX CPP 110 PCMCIA	La carte doit être installée dans l'emplacement PCMCIA type III du processeur.
	Configuration	• Paramétrage de configuration. • Déclaration de la configuration du bus par le logiciel **Sycon** et génération du fichier de configuration **.CO* . • Sélection du fichier de configuration **.CO* par l'outil Unity Pro.
Local ou connecté	Symbolisation	Symbolisation des variables associées à la carte CANopen.
	Programmation	Programmation des fonctions spécifiques : • des objets bit et des mots associés, • Instructions spécifiques.
En ligne	Transfert	Transfert de l'application vers l'automate. Un transfert de l'application vers l'automate ou un démarrage à froid du résultat de l'application démarre et configure la carte TSX CPP 110.
	Mise au point Diagnostics	Différents moyens sont disponibles pour la mise au point de l'application, le pilotage des entrées/sorties et le diagnostic des défaillances : • Objets langage ou IODDT, • Ecran de mise au point UnityPro, • Signalisation par voyants.
Local ou connecté	Documentation	Impression des différentes informations relatives à la configuration de la carte TSX CPP 110.

NOTE : L'ordre indiqué ci-dessus l'est uniquement à titre indicatif. Le logiciel Unity Pro rend possible l'utilisation des éditeurs dans l'ordre désiré et de manière interactive (cependant, les données ou l'éditeur de programme ne peuvent être utilisés sans avoir au préalable réalisé la configuration).

Méthode de mise en oeuvre

Vue d'ensemble

Le logigramme suivant présente la méthodologie de mise en œuvre d'une carte TSX CPP 110.


Configuration de la carte dans l'automate
Déclaration et configuration de la carte TSX CPP 110 et des esclaves par l'outil SyCon.
Configuration du comportement du bus
Sauvegarde du fichier de configuration .CO
Mode SyCon
Mode Unity Pro
Chargement de la configuration dans la carte par l'outil SyCon.
Sélection du fichier *.CO par l'outil Unity Pro.
Transfert l'application dans
Mise au point et diagnostics
Documentation

Recommandations

Lorsque la configuration du bus CANopen est trop importante pour permettre le changement vers le mode Unity Pro, vous devez réaliser les vérifications suivantes :

- Vérifier si vous pouvez rester en mode Unity Pro en utilisant un processeur Premium plus puissant *(voir page 30)*.
- S'il n'est pas possible de sélectionner un processeur plus puissant, passez en mode Sycon.

Adressage topologique CANopen

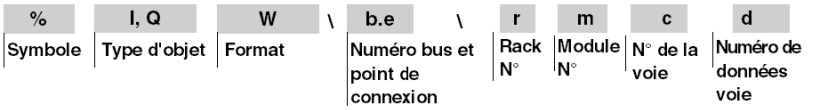
Présentation

L'adressage topologique (qui dépend de la position géographique de l'élément à adresser) est disponible sur les bus CANopen.

Toutefois, cela diffère légèrement de l'adressage topologique Fipio.

Illustration

L'adressage est défini de la manière suivante :



Syntaxe

Le tableau ci-dessous décrit les différents éléments constituant l'adressage.

Famille	Elément	Valeurs	Signification
Symbole	%	-	Indique un objet CEI
Type d'objet	I Q	- -	Objets d'entrée. Objets de sortie. Ces informations sont échangées de manière automatique à chaque cycle de la tâche à laquelle la carte CANopen est associée (MAST ou FAST).
Format (taille)	W	16 bits	Mot de 16 bits de type WORD.
Adresse module/voie et point de connexion	b	3 à 999	Numéro de bus.
	e	1 à 127	Numéro du point de connexion (numéro d'esclave CANopen).
Numéro du rack	r	0	Numéro du rack virtuel, toujours 0.
Numéro de module	m	0	Numéro de module virtuel, toujours 0.
N° de la voie	c	0	Numéro de voie virtuel, toujours 0.
Numéro de données voie	d	0 à 59	Numéro de donnée de l'esclave. Ce numéro est de maximum 59 car un esclave ne peut avoir qu'un maximum de 60 mots en entrées et en sorties (la somme de tous les mots d'entrée/sortie doit être inférieure à 60).

3.2 Configuration

Objet de cette section

Cette section décrit la configuration d'une carte PCMCIA TSX CPP 110.

Contenu de ce sous-chapitre

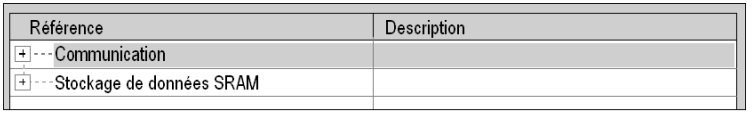
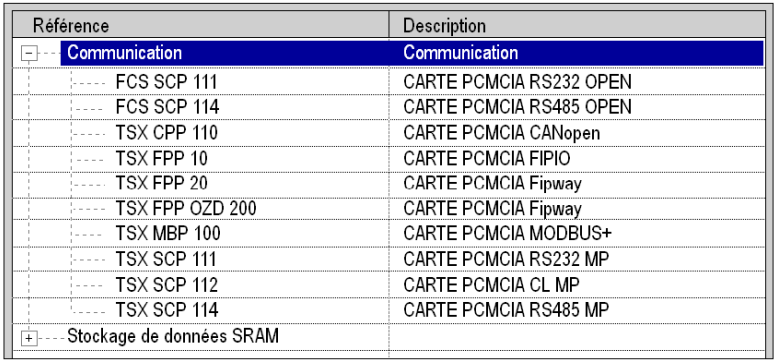
Ce sous-chapitre contient les sujets suivants :

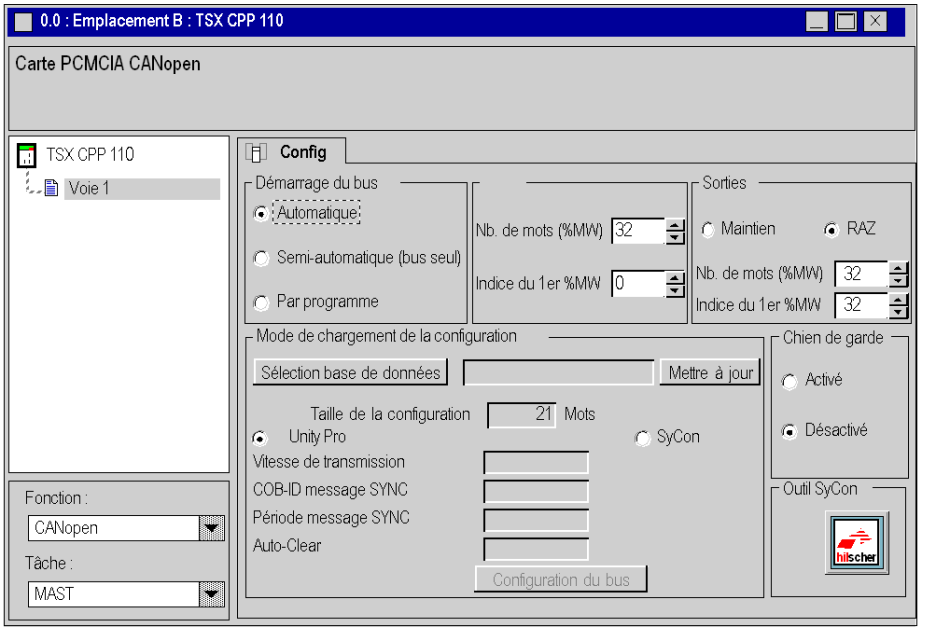
Sujet	Page
Accès à l'écran de configuration	37
Ecran de configuration de la carte TSX CPP 110 PCMCIA	39
Description des données d'E/S et du comportement du bus au démarrage	42
Procédure de sélection du mode de chargement de la configuration	45
Comment effectuer un chargement de configuration par X-Way	47
Ecran de configuration des esclaves	49
Procédure de configuration de la carte PCMCIA CANopen	50
Bus CANopen affiché dans le navigateur de projet	53
Fichier de configuration de la carte TSX CPP 110 PCMCIA	54

Accès à l'écran de configuration

Procédure

Cette opération permet de déclarer une carte TSX CPP 110 dans l'emplacement PCMCIA type III du processeur. L'exemple ci-dessous décrit la procédure.

Etape	Action
1	Ouvrez l'éditeur de configuration matérielle à partir du navigateur d'application.
2	Double-cliquez sur l'emplacement de la carte PCMCIA situé en bas du processeur. **Résultat :**la liste suivante apparaît. 
3	Ouvrez la ligne **Communication** en cliquant sur cette ligne **+** afin d'obtenir la liste des cartes de communication disponibles pour cet emplacement. **Résultat :** la liste suivante apparaît. 
4	Sélectionnez la carte TSX CPP 110 puis validez par **OK**. **Résultat** : le logiciel affiche le nouvel éditeur de configuration X-Bus et vous pouvez maintenant accéder à l'écran de configuration de la carte PCMCIA.

Etape	Action
5	Double-cliquez sur l'emplacement de la carte PCMCIA pour obtenir la fenêtre de configuration de la carte PCMCIA. **Résultat :** 

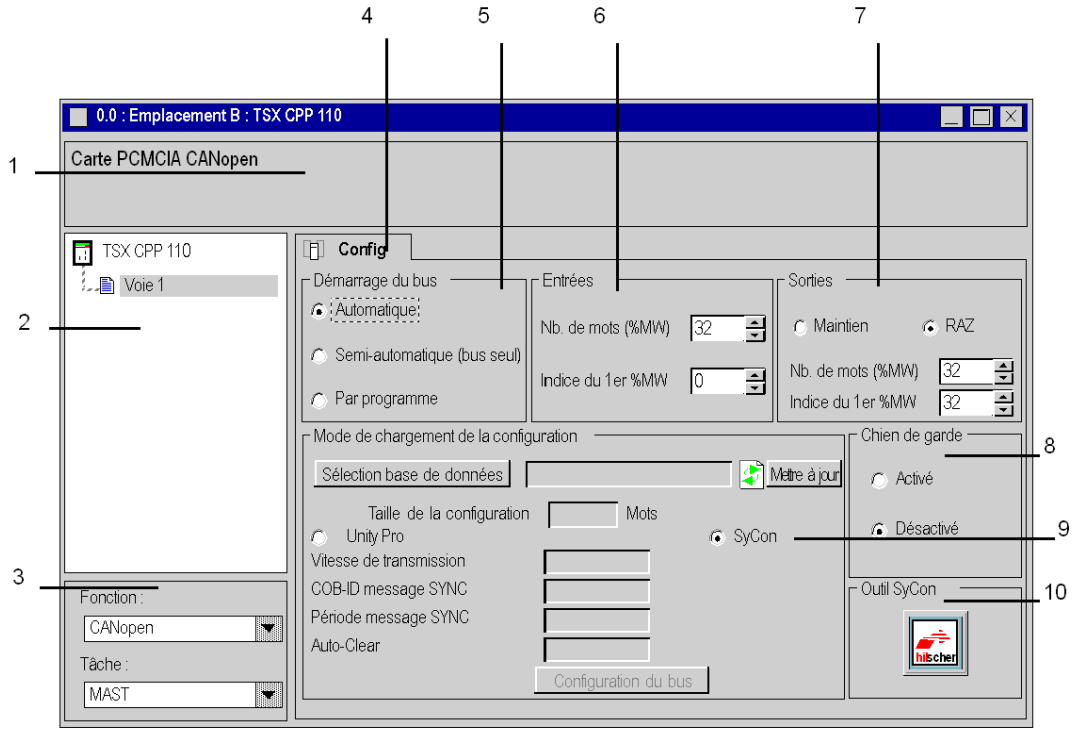
Ecran de configuration de la carte TSX CPP 110 PCMCIA

Vue d'ensemble

Cet écran permet de déclarer la voie de communication et de configurer les paramètres nécessaires à une liaison CANopen.

Illustration

L'écran de communication CANopen se présente comme suit :



Eléments et fonctions

Ce tableau décrit les différentes zones constituant l'écran de configuration :

Zone	Nb	Fonction
Module	1	Ce champ comprend l'intitulé abrégé de la carte PCMCIA.
Voie	2	Cette zone permet de sélectionner la voie de communication à configurer. Cliquez sur la référence de la carte afin d'afficher les onglets : ● **Description** qui donne les caractéristiques de la carte de communication. ● **Objets d'E/S** *(voir Unity Pro, Modes de marche)*, qui permet de présymboliser les objets d'entrée/de sortie ● **Défaut** qui donne accès aux défauts de la carte (en mode connecté uniquement).
Paramètres généraux	3	Cette zone est composée des éléments suivants : ● le choix des fonctions de communication, dans ce cas CANopen, ● une liste déroulante permettant d'associer le bus CANopen à la tâche d'une application. Cette liste déroulante comprend trois options qui règlent la fréquence de mise à jour des zones de stockage associées aux E/S : ● MAST : Fréquence des tâches MAST ● FAST : Fréquence des tâches FAST
Tab	4	L'onglet au premier plan indique le type d'écran affiché. Dans notre cas il s'agit de l'écran de configuration.
Config	5	Cette zone permet de sélectionner le comportement du bus au démarrage.
	6	Cette zone permet de configurer l'adresse (mémoire interne de l'automate) ou seront copiées périodiquement les entrées des équipements CANopen.
	7	Cette zone permet de configurer le mode de repli des sorties des équipements du bus ainsi que l'adresse (mémoire interne de l'automate) ou seront lues périodiquement les sorties des équipements CANopen.
	8	Cette zone permet d'activer ou de désactiver le chien de garde du bus CANopen. Le chien de garde est activé par défaut. Il est déclenché lorsque la carte PCMCIA ne peut plus gérer correctement le bus. Lorsqu'il est déclenché, il fait passer les sorties des esclaves à zéro.
	9	Cette zone permet de configurer le bus : ● Sélection du fichier de configuration Sycon (*.CO). *(voir page 50)* ● Configuration Unity Pro ou Sycon *(voir page 45)*
	10	Ce bouton permet de lancer le logiciel **Sycon** s'il est installé sur le PC.

⚠ AVERTISSEMENT

COMPORTEMENT INATTENDU DE L'APPLICATION

Avant de désactiver le chien de garde, assurez-vous que la carte PCMCIA ne gère pas le bus CANOpen, et que le comportement des équipements reste acceptable.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

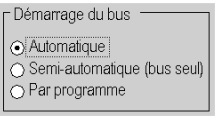
Description des données d'E/S et du comportement du bus au démarrage

Présentation

L'écran de configuration permet de configurer le comportement du bus au démarrage ainsi que les entrées et les sorties des équipements esclaves sur le bus.

Démarrage du bus

La figure ci-dessous illustre la zone de configuration du démarrage du bus.



Le démarrage du bus peut être opéré de trois façons :

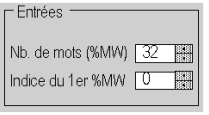
- **Automatique** : la configuration du bus, la gestion de la communication et la mise à jour des E/S esclaves sont activées au démarrage, sans intervention d'une application.
- **Semi-automatique (bus seul)** : la configuration du bus, la gestion de la communication sont activées au démarrage mais la gestion des E/S doit être confirmée par l'application utilisant les objets langage *(voir page 95)* correspondants.
- **Par programme** : le démarrage du bus doit être entièrement géré par l'application utilisant les objets langage *(voir page 95)* correspondants.

NOTE : en mode**Automatique** et **Semi-automatique** si l'option **"Arrêt du bus lorsque le chien de garde esclave est déclenché"** est activée, en cas de défaut, vous devez opérer un démarrage à froid de l'application de l'automate pour redémarrer le bus CANopen.

NOTE : que le mode démarrage soit configuré ou non, l'option **"Arrêt du bus lorsque le chien de garde esclave est déclenché"** arrête le bus lorsqu'un esclave disparaît. Si un défaut survient avant le démarrage du bus, le démarrage est néanmoins opéré.

Entrées

La figure ci-dessous illustre la zone de configuration du démarrage des entrées.



Pour configurer les entrées des esclaves du bus il est nécessaire d'indiquer une zone mémoire dans laquelle elles seront copiées périodiquement. Pour définir cette zone, indiquez :

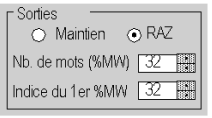
- **Un nombre de mots** : il s'agit du nombre de mots d'entrée correspondant à la taille des données d'entrée configurées par le logiciel **Sycon**.
- **Adresse du premier %MW** : il s'agit de l'adresse du premier mot dans la zone de mémoire d'entrée.

NOTE : Les mots `%MW` contiennent les valeurs d'entrée des esclaves du bus. Lorsque le mode de chargement de Unity Pro est choisi, il est possible de reconnaître les mots associés aux esclaves en appuyant sur le bouton **Configuration du bus** sur l'écran de configuration. Dans le mode de chargement Sycon, seul le logiciel Sycon vous permet de reconnaître les esclaves associés aux mots `%MW`. Ces mots `%MW` sont utilisés directement comme entrées par l'application.

NOTE : Comme pour le bus Fipio, si le mot`%SW8` est utilisé (`le bit %SW8.0` paramétré sur 1 pour la tâche maître et `le bit %SW8.1` paramétré sur 1 pour la tâche rapide), la phase d'acquisition d'entrées des équipements du bus est inhibée. Les valeurs de ces entrées conservent l'état dans lequel elles se trouvaient avant le paramétrage du bit sur 1.

Sorties

La figure ci-dessous illustre la zone de configuration des sorties.



Pour configurer les sorties, il est nécessaire d'indiquer, comme pour les entrées, la table de mots qui contiendra la valeur des sorties du bus mais aussi le type de repli nécessaire lorsqu'une défaillance d'esclave survient :

- Maintien
- Réinitialisation.

NOTE : Les mots `%MW` contiennent les valeurs de sortie des esclaves du bus. Lorsque le mode de chargement de Unity Pro est choisi, il est possible de reconnaître les mots associés aux esclaves en appuyant sur le bouton **Configuration du bus** sur l'écran de configuration. Dans le mode de chargement Sycon, seul le logiciel Sycon vous permet de reconnaître les esclaves associés aux mots `%MW`. Ces mots `%MW` sont utilisés directement comme sorties par l'application.

NOTE : Les tables de mots sont situées dans la mémoire interne de l'automate. Il est interdit de faire chevaucher ces deux zones.

NOTE : Si le nombre de mots d'entrées et de mots de sorties diffère de celui déterminé dans le fichier de configuration *(voir page 45)* (nom de fichier **.CO*), le logiciel Unity Pro le signale lorsque la configuration est confirmée.

NOTE : La taille maximale autorisée de la zone de mémoire réservée aux E/S dépend du type de processeur et de la tâche associée *(voir page 30)*.

NOTE : Comme pour le bus Fipio, si le mot`%SW9` est utilisé (`le bit %SW9.0` paramétré sur 1 pour la tâche maître et `le bit %SW9.1` paramétré sur 1 pour la tâche rapide), les sorties des équipements sur le bus sont maintenues dans l'état dans lequel elles se trouvaient avant le paramétrage du bit sur 1.

Procédure de sélection du mode de chargement de la configuration

Présentation

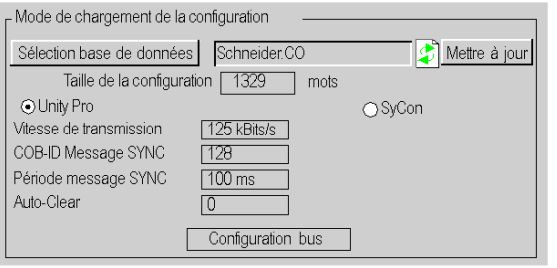
L'écran de configuration permet de sélection le mode de chargement de la configuration. Les deux choix possibles sont les suivants :

- Chargement à l'aide du logiciel Unity Pro,
- Chargement à l'aide du logiciel Sycon.

Dans les deux cas de figure, il est essentiel de sélectionner une base de données de configuration *(voir page 50)* créée avec le logiciel Sycon.

Illustration

La figure ci-dessous montre la zone de l'écran de configuration qui permet de sélectionner le mode de chargement de la configuration.



Description

Le tableau ci-dessous présente les différents choix possibles.

Zone	Description
Sélection base de données	Cette zone permet de sélectionner la base de données qui correspond à la configuration du bus géré par la carte PCMCIA TSX CPP 110. Cette configuration est réalisée par le logiciel Sycon qui génère le fichier **.CO* qui doit être sélectionné *(voir page 50)*.
Mettre à jour	Lorsque vous cliquez sur ce bouton, le fichier **.CO* sélectionné est rechargé et évalué. **Remarque** : Cette manœuvre doit être réalisée après chaque modification faite par Sycon au fichier **.CO* sélectionné.
Unity Pro	Lorsque vous sélectionnez ce bouton, la configuration du bus est chargée avec l'application de l'automate. Lorsque l'application est trop grande (taille de mémoire plus grande que celle autorisée pour le processeur), le logiciel Unity Pro n'autorise pas cette sélection et vous devez sélectionner **Sycon** comme expliqué ci-dessous.

Zone	Description
Sycon	Lorsque vous sélectionnez ce bouton, la configuration du bus est considérée comme déjà chargée dans la carte PCMCIA à l'aide du PC et du logiciel Sycon. Le logiciel Unity Pro réalise simplement une vérification pour s'assurer que la configuration de la carte est identique à celle du fichier **.CO* sélectionné.
Vitesse de transmission	Lorsque le mode de chargement Unity Pro a été sélectionné, cette zone affiche la vitesse de transmission sur le bus défini sous Sycon.
COB-ID message SYNC	Lorsque le mode de chargement Unity Pro a été sélectionné, cette zone affiche le COB-ID message SYNC sélectionné dans Sycon.
Période message SYNC	Lorsque le mode de chargement Unity Pro a été sélectionné, cette zone affiche la période du bus défini sous Sycon.
Auto-Clear	Lorsque le mode de chargement Unity Pro a été sélectionné, cette zone affiche le mode **Auto-clear on** ou **Auto-clear off** sélectionné sous Sycon.
Configuration bus	Lorsque le mode de chargement Unity Pro a été sélectionné, ce bouton permet d'accéder à la configuration des esclaves du bus.

Comment effectuer un chargement de configuration par X-Way

Généralités

Lorsque le bus CANopen est configuré en mode de chargement par Sycon, il est possible de télécharger la configuration de la carte TSX CPP 110 grâce au Driver X-Way.

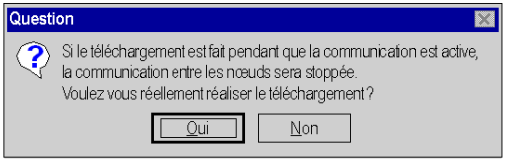
Ce téléchargement peut s'effectuer au travers d'un réseau Ethernet, ou tout simplement sur un bus Uni-Telway.

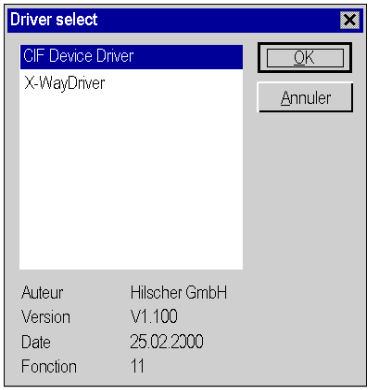
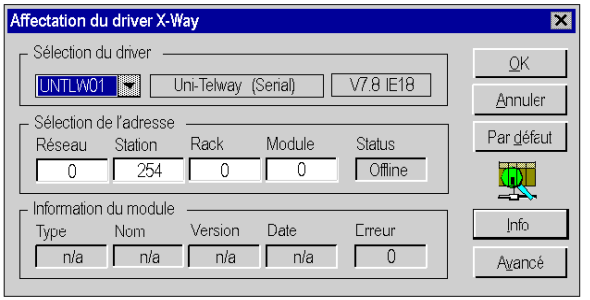
NOTE : ATTENTION à bien mettre l'automate en STOP lors de la procédure de téléchargement.

NOTE : lorsque le choix du driver de communication X-Way a été validé, vous êtes obligé de quitter le logiciel Sycon pour changer de driver.

Marche à suivre

Ce tableau décrit les étapes à effectuer pour charger la configuration d'une carte CANopen grâce aux drivers de communication X-Way.

Etape	Action
1	Connectez-vous sur l'automate qui contient la carte TSX CPP 110 à l'aide du logiciel Unity Pro.
2	Passez cet automate en STOP.
3	Lancez le logiciel Sycon.
4	Chargez ou créez la configuration désirée à l'aide du logiciel Sycon.
5	Sélectionnez la commande **En ligne →Télécharger**. **Résultat** : un message indiquant que lors du chargement de la configuration la communication entre les esclaves est stoppée, apparaît. 

Etape	Action
6	Cliquez sur **Oui** pour indiquer que vous êtes d'accord avec cet arrêt des communications inter-esclaves. **Résultat** : une fenêtre de sélection du driver **X-Way** ou **CIF** apparaît. 
7	Sélectionnez le driver **X-Way** puis cliquez sur **OK**. **Résultat** : la fenêtre **Affectation du driver X-Way** apparaît. 
8	Sélectionnez le driver désiré (Uni-Telway, XIP,...) à partir de la zone **Sélection du driver**.
9	Saisissez l'adresse de l'automate (Réseau, Station, Rack, Module) puis cliquez sur **OK** pour lancer le téléchargement. **Résultat** : tant que le téléchargement est en cours, une fenêtre indique l'avancement du transfert des données. Lorsque le transfert est terminé, cette fenêtre disparait et laisse la place à l'écran principal de configuration du bus.

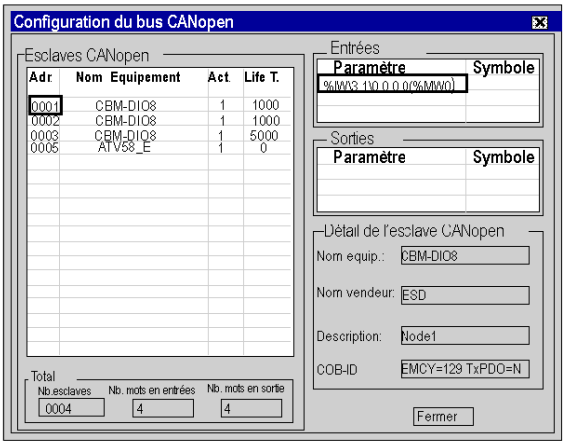
Ecran de configuration des esclaves

Présentation

Avec le logiciel Unity Pro, il est possible d'accéder à la configuration des esclaves du bus. Les informations contenues dans l'écran sont sensiblement identiques à celles de l'écran de mise au point *(voir page 67)*.

Illustration

La figure suivante présente l'écran décrivant la configuration des esclaves.



Fonctionnement

- Cliquez sur un esclave parmi la liste des esclaves CANopen.
- Les paramètres de l'esclave, en entrées et en sorties apparaissent alors dans les zones **Entrées** et **Sorties**. Pour plus de clarté, l'affichage de l'adressage topologique CANopen *(voir page 35)* et de la zone mémoire réservée dans l'automate est proposé.
- Les informations sur l'esclave apparaissent dans la zone **Détail de l'esclave CANopen**.

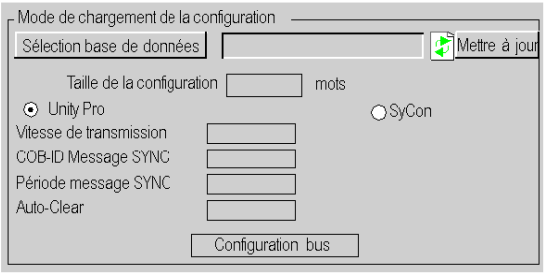
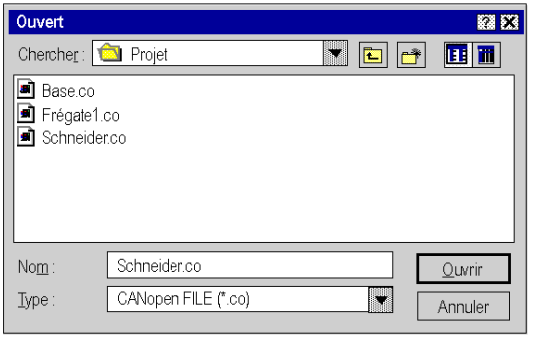
Procédure de configuration de la carte PCMCIA CANopen

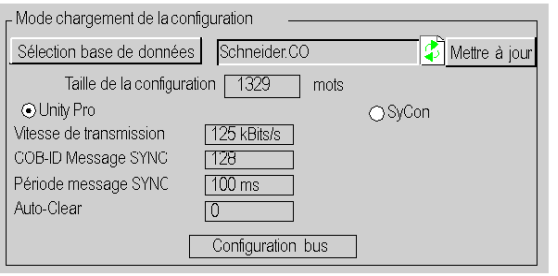
Présentation

Pour configurer la carte TSX CPP 110, certaines procédures sont essentielles ou nécessitent une explication spéciale. Des détails sont disponibles pour les procédures suivantes :

Procédure de sélection d'un fichier de configuration

Le tableau ci-dessous décrit les étapes de la procédure de sélection d'une configuration CANopen.

Etape	Action
1	Cliquez sur **Sélection base de données** :  **Résultat** : l'écran suivant apparaît : 

Etape	Action
2	Sélectionnez le fichier *.*CO* requis puis cliquez sur **OK**. **Résultat** : si le nombre de mots réservés aux entrées et aux sorties correspond à la configuration sélectionnée, la configuration apparaît dans l'écran de configuration Unity Pro.  Dans le cas contraire, les valeurs incorrectes apparaissent en rouge dans l'écran de configuration et il est impossible d'accéder à la configuration du bus tant que ces valeurs ne sont pas rectifiées. Lorsque le fichier *.CO* dépasse la capacité maximale de données de configuration *(voir page 30)*, un message d'erreur apparaît. Vous devez soit utiliser un processeur Premium plus puissant, soit passer en mode **Sycon** et charger la configuration directement sur la carte utilisant le logiciel **Sycon**.

Procédure de configuration d'une carte CANopen

Le tableau suivant décrit la procédure de configuration d'une carte PCMCIA CANopen TSX CPP 110.

Etape	Action
1	Sélectionnez le type de démarrage de bus.
2	Cliquez sur le bouton **Outil Sycon** pour démarrer le logiciel de configuration **Sycon**.
3	A l'aide du logiciel **Sycon**, configurez votre bus CANopen en fonction des équipements prévus pour le bus.
4	Sauvegarde de la configuration dans un fichier avec une extension .CO.
5	Revenez dans Unity Pro

Etape	Action
6	Sélection d'un fichier de configuration *(voir page 50)*.
7	Réservez les zones de mémoire automate à associer aux entrées et aux sorties.
8	Sélectionnez la case d'option Unity Pro si vous souhaitez incorporer la configuration CANopen dans l'application Unity Pro.
9	Cliquez sur le bouton Chien de garde **Activé**.
10	Validez la configuration. **Résultat** : le logiciel vous indique le nombre de mots dont vous avez besoin pour l'entrée et la sortie pour configurer le bus sélectionné. Si les tailles spécifiées ne sont pas les tailles optimales, vous pouvez les ajuster.

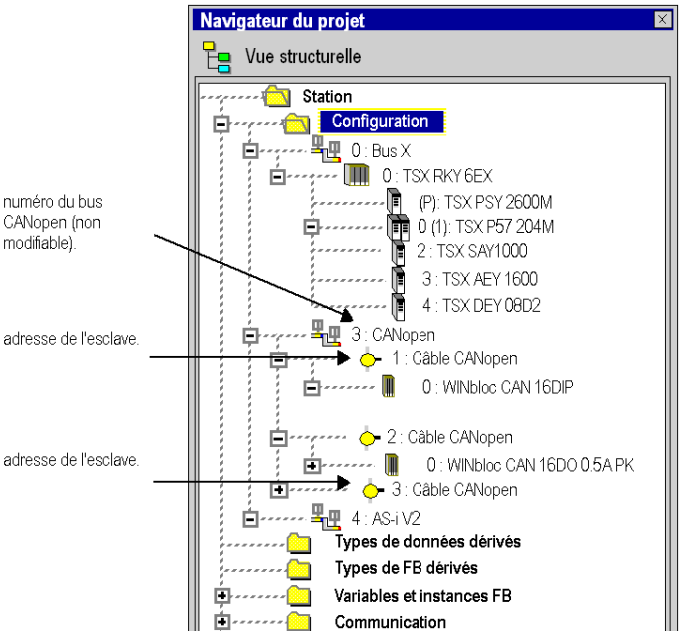
Bus CANopen affiché dans le navigateur de projet

Présentation

Lorsque vous déclarez une carte TSX CPP 110 dans un processeur d'automate, le bus CANopen est représenté dans le répertoire *Configuration* du navigateur projet. Le numéro du bus CANopen est calculé automatiquement par Unity Pro. Cette valeur n'est pas modifiable.

Après avoir chargé le fichier de configuration **.CO* créé à l'aide du logiciel Sycon et validé la configuration, les esclaves CANopen apparaissent sur le bus CANopen du navigateur de projet. Chaque esclave est accompagné de son numéro d'adresse. L'affichage du bus CANopen et des esclaves vous permet de voir leur adressage topologique.

La figure suivante présente le bus CANopen avec les esclaves dans le navigateur du projet.



Fichier de configuration de la carte TSX CPP 110 PCMCIA

Présentation

A partir du module processeur, un dossier décrivant la configuration de l'application pour la carte PCMCIA TSX CPP 110 est disponible dans l'éditeur de documentation.

Illustration

La représentation est la suivante :

	TSX P57 354M [RACK 0 POSITION 0]		
Identifiant module			
Référence du produit :	TSX P57 354M	**Désignation :**	PROCESSEUR TSX P57 354M
Adresse :	000	**Symbole :**	
Paramètres voie : 0			
Affectation tâche/voie :	MAST		
Type de voie :	Port terminal	Symbole voie :	
Fonction métier :	LIAISON Uni-Telway	Symbole voie :	
Vitesse de transmission	19 200 bits/s	Temporisation :	30 ms
Type de module :	Maître	Parité :	impaire
Nombre d'esclaves :	8		
Paramètres voie : 1			
Affectation tâche/voie :	MAST		
Type de sous-module :	CARTE PCMCIA CANopen TSX CPP 110		
Type de voie :	Port PCMCIA	Symbole voie :	
Fonction métier :	CANopen		
Entrées	Adresse du 1er %MW	32	Longueur : 424
Sorties	Adresse du 1er %MW	1056	Longueur : 102
	Stratégie de repli Sortie : Remise à 0		

Mode de configuration :	Automatique	Chien de garde TSX CPP 110 :		actif
Mode chargement :	Unity Pro	Fichier de configuration CANopen :		E:\BG1.CO
Configuration du bus CANopen :	Vitesse de transmission :	1 Mbits/s	COB-ID message Sync :	128
	Auto-Clear :	éteint	Période message sync :	100 ms

Configuration de l'esclave CANopen

Adresse	Type	Act./Guard poll
1	AMM 09000.	1 / 1
2	Profil 401 standard EDS	1 / 0
3	ADM 37010	1 / 0

Objets langage d'esclave CANopen :

Adresse	Entrées	Symbole	Sorties	Symbole
1.	%MW32		%MW1056	
	%MW33			
	%MW34			
	%MW35		%MW1057	
	%MW36		%MW1058	

3.3 Programmation

Objet de cette section

Cette section décrit les outils disponibles pour programmer le fonctionnement d'un bus CANopen géré par une carte PCMCIA TSX CPP 110 et obtenir des informations sur celui-ci.

Il est possible de programmer le fonctionnement du bus CANopen à l'aide des requêtes UNI-TE :

- Envoyez et recevez des messages SDO sur le bus,
- Accédez à la couche de liaison en envoyant des PDU.

Il est également possible de surveiller le bus et son fonctionnement :

- Identification du maître,
- Envoi de requêtes de diagnostics sur les équipements du bus.

Ces requêtes sont envoyées au maître CANopen (carte PCMCIA TSX CPP 110) pour être traitées.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Accès aux SDO CANopen	57
Demande IDENTIFICATION	63
Commandes diagnostics	65

Accès aux SDO CANopen

Présentation

Les fonctions de communication READ_VAR *(voir Unity Pro, Communication, Bibliothèque de blocs)* et WRITE_VAR *(voir Unity Pro, Communication, Bibliothèque de blocs)* sont utilisées pour accéder au transfert des données CANopen de type SDO. Les paramètres de ces fonctions déterminent l'action exécutée.

Ces services sont basés sur le traitement normalisé des messages CMS de la norme CANopen. Consultez la documentation consacrée aux esclaves CANopen pour plus d'informations sur les formats SDO utilisés.

⚠ ATTENTION

COMPORTEMENT INATTENDU DE L'APPLICATION

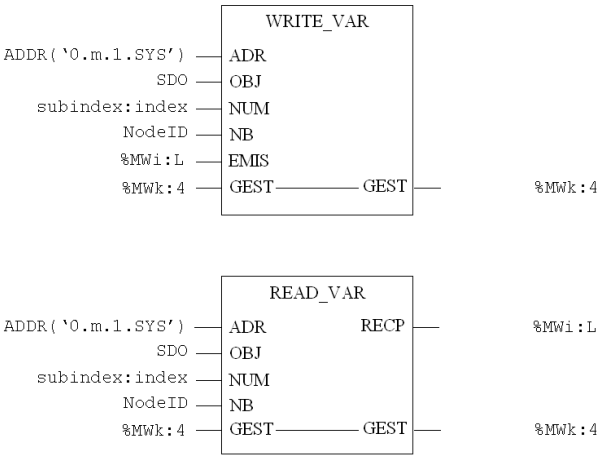
N'envoyez pas ou ne recevez pas de SDO simultanément.

Le non-respect de ces instructions peut provoquer des blessures ou des dommages matériels.

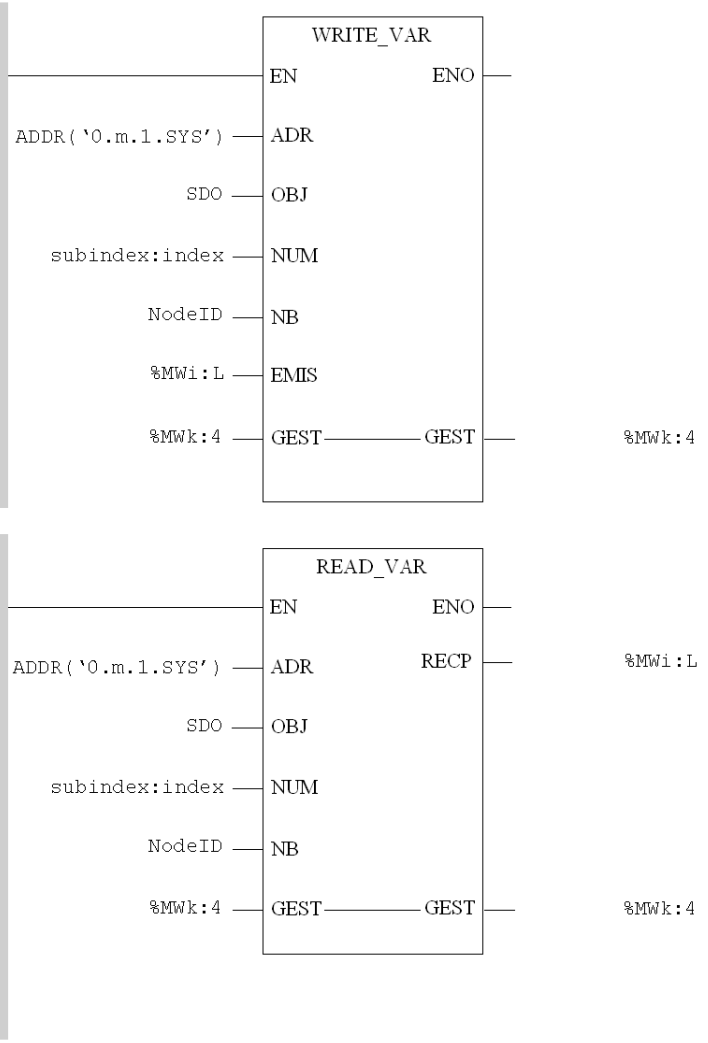
Si des SDO sont envoyés et reçus simultanément, le SDO suivant peut ne pas être traité. Dans ce cas, effectuez un redémarrage à froid à l'aide du bouton RAZ du processeur pour réinitialiser l'application et revenir en mode de fonctionnement normal.

Représentation

Représentation en FBD :



Représentation en LD :



Représentation en ST :

La fonction de communication a la syntaxe suivante :

```
WRITE_VAR(ADDR('0.m.1.SYS'), 'SDO', index:subindex, NodeID,
%MWi:L, %MWk:4)
```

```
READ_VAR(ADDR('0.m.1.SYS'), 'SDO', index:subindex, NodeID,
%MWk:4, %MWi:L)
```

Description des paramètres de la fonction WRITE_VAR

La fonction de communication a la syntaxe suivante :

Le tableau suivant décrit les différents paramètres de la fonction.

Paramètre	Description
`ADDR('0.m.1.SYS')`	Adresse de l'entité destinataire de l'échange : ● m : emplacement du processeur dans le rack (0 ou 1) ● 1: voie (toujours 1) ● SYS : serveur UNI-TE de la carte PCMCIA
`'SDO'`	Type d'objet SDO (**SDO** toujours en majuscules)
`subindex:index`	Double mot ou valeur immédiate identifiant l'index et le sous-index SDO CANopen : Le mot de poids fort composant le mot double contient le sous-index et le mot de poids faible l'index. **Exemple** : si vous utilisez le mot double %MD0 : ● %MW0: contient l'index, ● %MW1: contient le sous-index.
`NodeID`	Mot ou valeur identifiant l'équipement destinataire sur le bus CANopen
`%MWi:L`	Table de mots contenant les données à envoyer (longueur minimum = 1)
`%MWk:4`	Paramètres de gestion des échanges : quatre mots identifiant l'adresse des données utilisée pour contrôler la fonction appelée `WRITE_VAR`

Description des paramètres de la fonction READ_VAR

La fonction de communication a la syntaxe suivante :

Le tableau suivant décrit les différents paramètres de la fonction.

Paramètre	Description
`ADDR('0.m.1.SYS')`	Adresse de l'entité destinataire de l'échange : ● m : emplacement du processeur dans le rack (0 ou 1) ● 1 : voie (toujours 1) ● SYS : serveur UNI-TE de la carte PCMCIA
`'SDO'`	Type d'objet SDO (**SDO** toujours en majuscules)
`subindex:index`	Double mot ou valeur immédiate identifiant l'index et le sous-index SDO CANopen : Le mot de poids fort composant le mot double contient le sous-index et le mot de poids faible l'index. **Exemple** : si vous utilisez le mot double %MD0 : ● %MW0: contient l'index, ● %MW1: contient le sous-index.

Paramètre	Description
NodeID	Mot ou valeur identifiant l'équipement destinataire sur le bus CANopen
%MWk:4	Paramètres de gestion des échanges : quatre mots identifiant l'adresse des données utilisée pour contrôler la fonction appelée READ_VAR
%MWi:L	Table de mots contenant les données à recevoir (longueur minimum = 1)

Paramètres de gestion

Les paramètres de gestion sont regroupées dans un tableau de 4 nombres entiers. Les valeurs contenues dans ce tableau permet de gérer les fonctions de communication. Elles sont détectées par le processeur qui a exécuté la fonction.

Les mots %MWk sont détaillés dans le tableau suivant :

Numéro du mot	Octet de poids fort	Octet de poids faible	Données gérées par
%MWk	Numéro d'échange	Bit d'activité	Système
%MWk+1	Rapport d'opération	Rapport de communication	
%MWk+2	Timeout		Vous
%MWk+3	Longueur d'octet : • dans le cas d'une fonction WRITE_VAR, initialisez ce mot avec le nombre d'octets à envoyer. • dans le cas d'une fonction READ_VAR, lorsque la requête est terminée, ce mot contient le nombre de caractères reçus dans la table de mots des données reçues.		

NOTE : une fonction peut détecter une erreur de paramètre avant d'activer l'échange. Le bit d'activité reste à 0 et le rapport de communication est initialisé avec les valeurs correspondant à l'erreur détectée.

Le tableau suivant explique la signification des valeurs du mot %MWk+1 :

Numéro du mot	Rapport d'opération	Rapport de communication
%MWk+1	Ces codes sont destinés aux services CANopen sur une application distante. 16#00 : Résultat positif 16#01 : Requête non traitée 16#02 : Réponse incorrecte 16#03 : Réservé	Ces codes sont destinés aux blocs fonction READ_VAR, WRITE_VAR et SEND_REQ sur une application distante. Ce rapport n'est pertinent que lorsque le bit d'activité passe de 1 à 0. 16#00 : Echange de messages correct
	16#00	16#01: Echange arrêté par un timeout 16#02 : Echange arrêté par une demande de l'utilisateur (ANNULER) 16#03 : Format d'adresse incorrect 16#04 : Format de destination incorrect 16#05 : Format du paramètre de gestion incorrect 16#06 : Paramètre spécifique incorrect 16#07 : Problème d'émission vers le destinataire 16#08 : Réservé 16#09 : Taille du tampon de réception insuffisante 16#0A : Taille du tampon d'envoi insuffisante 16#0B : Pas de ressources système du processeur 16#0C : Numéro d'échange incorrect 16#0D : Aucun télégramme reçu 16#0E : Longueur incorrecte 16#0F : Service télégramme non configuré 16#10 : Module réseau absent 16#11 : Requête manquante 16#12 : Serveur d'application déjà actif 16#13 : Numéro de transaction UNI-TE incorrect

Numéro du mot	Rapport d'opération	Rapport de communication
%MWk+1	16#01 : Pas de ressources vers le processeur 16#02 : Pas de ressources de ligne 15#03 : Aucun équipement ou équipement sans ressource **Remarque** Ce code n'est géré que par les cartes PCMCIA TSX FPP 10 et TSX FPP 20. 16#04 : Erreur de ligne détectée 16#05 : Erreur de longueur détectée 16#06 : Voie de communication inopérante 16#07 : Erreur d'adressage détectée 16#08 : Erreur d'application détectée 16#0B : Pas de ressources système 16#0C : Fonction de communication non active 16#0D : Destinataire absent 16#0F : Problème de routage interstation ou voie non configurée 16#11 : Format d'adresse incorrect 16#12 : Pas de ressource de destination 16#14 : Connexion inopérante (exemple : Ethernet TCP/IP) 16#15 : Aucune ressource sur la voie locale 16#16 : Accès non autorisé (exemple : Ethernet TCP/IP) 16#17 : Configuration réseau incorrecte (exemple : Ethernet TCP/IP) 16#18 : Connexion temporairement indisponible 16#21 : Serveur d'application arrêté 16#30 : Erreur de transmission détectée	16#FF : Message refusé

Demande IDENTIFICATION

Présentation

Cette demande permet d'identifier le maître du bus CANopen (carte PCMCIA TSX CPP 110).

Cette demande est réalisée à l'aide de la fonction de communication SEND_REQ *(voir Unity Pro, Communication, Bibliothèque de blocs)*.

Syntaxe

La fonction de communication a la syntaxe suivante :

```
SEND_REQ(ADDR('0.m.1.SYS'), 16#0F, %MWi:L, %MWk:4, %MWj:L)
```

Le tableau suivant décrit les différents paramètres de la fonction.

Paramètre	Description
ADDR('0.m.1.SYS')	Adresse de l'entité destinataire de l'échange. ● m : Numéro de l'emplacement du processeur, 0 ou 1 ● 1: voie (toujours 1) ● SYS : accès serveur UNI-TE de la carte PCMCIA
16#0F	Code de requête
%MWi:L	N'est pas utilisé pour la fonction IDENTIFICATION (la longueur est de 1)
%MWk:4	Paramètres de gestion des échanges : quatre mots identifiant l'adresse des données utilisée pour contrôler la fonction appelée IDENTIFICATION
%MWj:L	Table de mots contenant les informations d'identification de la carte. La longueur L doit être de 12.

%MWj:L

Le tableau suivant donne les détails des mots %MWj:L.

Nombre de mots	Octet de poids fort	Octet de poids faible
%MWj	Code produit : ● 16#05: Premium	16#FF
%MWj+1	Longueur de la chaîne d'identification : 16#0C	Numéro de la version codée BCD (Version codée 1.0 16#10)
%MWj+2	'S'	'T'
%MWj+3	' '	'X'
%MWj+4	'P'	'C'
%MWj+5	' '	'P'
%MWj+6	'1'	'1'

Nombre de mots	Octet de poids fort	Octet de poids faible
%MWj+7	16#00	'0'
%MWj+8	Etat du voyant. Le voyant COM est codé sur les deux premiers bits et le voyant ERR sur les deux suivants selon les séquences suivantes : ● 0, 0: Eteint, ● 0, 1: Clignotant, ● 1, 0: Allumé en permanence.	Etat de la carte PCMCIA : ● 0: Absent, ● 1: Autotest, ● 2: Erreur, ● 3: Prêt, ● 4: En attente, ● 5: Non configuré.
%MWj+9	Type de produit : 16#02	Type fonctionnel : 16#2E
%MWj+10	Type d'erreur ● bit 0 : Carte en mode test/mise au point, ● bit 1 : Erreur bus ou bus inactif, ● bit 2 : Absence d'unité de connexion, ● bit 3 : Carte auto-test ou inaccessible, ● bit 4 : Réservé, ● bit 5 : Carte différente de celle configurée, ● bit 6 : Carte absente, ● bit 7 : Erreur sur au moins un esclave.	Référence catalogue : 16#01
%MWj+11	-	16#00

Paramètres de gestion

Le tableau suivant donne les détails des mots `%MWk:4`.

Nombre de mots	Octet de poids fort	Octet de poids faible	Données gérées par
%MWk	Numéro d'échange	Bit d'activité	Le système
%MWk+1	Rapport d'activité, rapport 16#3F positif	Compte rendu de communication	
%MWk+2	Temporisation		Vous
%MWk+3	Longueur : l'initialisation à 0 est obligatoire avant l'envoi de la fonction		

Commandes diagnostics

Présentation

Les commandes diagnostics sont envoyées par le bloc fonction SEND_REQ *(voir Unity Pro, Communication, Bibliothèque de blocs)*. Dans ce cas, la fonction SEND_REQ permet :

- d'obtenir des diagnostics sur les esclaves du bus,
- d'obtenir la version d'une carte PCMCIA CAN open,
- d'obtenir des variables d'état,
- d'obtenir l'historique des erreurs du bus.

Syntaxe

La syntaxe de la fonction communication est la suivante :

```
SEND_REQ (ADDR('0.m.1.SYS', 16#0031, %MWi:3, %Mwk:4, %MWj:L)
```

Le tableau suivant décrit les différents paramètres de la fonction.

Paramètre	Description
ADDR('0.m.1.SYS')	Adresse de l'entité destinataire de l'échange. ● m : emplacement du processeur dans le rack (0 ou 1) ● 1: voie (toujours 1) ● SYS : serveur UNI-TE de la carte PCMCIA
16#0031	Code de requête

%MWi:3	Paramètres de la requête :
	● %MWi: Type objet diagnostic :
	  ● 1 ... 127: Diagnostics des esclaves 1 à 127
	  ● 128: Version carte CANopen
	  ● 129: Statut de la carte CANopen, la réponse est équivalente aux contenus des mots d'état %IW0.m.1.i *(voir page 92)*.
	  ● 130: Historique des messages d'erreur.
	● %MWi+1: Adresse de démarrage dans le tableau de diagnostic (valeur par défaut 0). Pour permettre un accès partiel au tableau de diagnostic, spécifiez un mot de démarrage dans le tableau (offset de démarrage)
	● %MWi+2 : Longueur des bits des diagnostics à lire, cette longueur est généralement deux fois supérieure à la longueur du tableau de réponse

%MWk:4	Paramètres de gestion des échanges : quatre mots identifiant l'adresse des données utilisée pour contrôler la fonction appelée SEND_REQ.
%MWj:L	Le tableau de réception contient les données de diagnostic *(voir page 71)*.

Paramètres de gestion

Le tableau suivant donne les détails des mots %MWk:4.

Nombre de mots	Octet de poids fort	Octet de poids faible	Données gérées par
%MWk	Numéro d'échange	Bit d'activité	Le système
%MWk+1	Compte rendu d'activité : • réponse positive : 16#61 • réponse incorrecte : 16#FD	Compte rendu de communication	
%MWk+2	Temporisation		Vous
%MWk+3	Longueur : nombre d'octets de réponses (initialisation obligatoire pour activer la fonction si le nombre d'octets envoyés est de 6)		

3.4 Mise au point

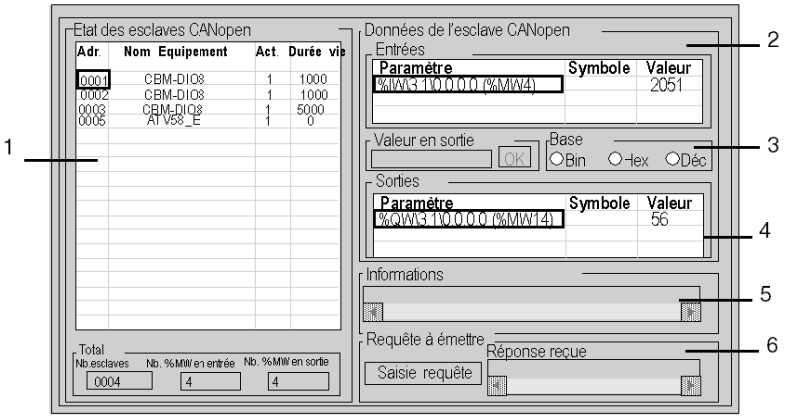
Description de l'écran de mise au point

Présentation

La fonction Mise au point ou le double clic sur la carte PCMCIA TSX CPP 110 sur la configuration du logiciel Unity Pro est disponible en mode connecté uniquement.

Illustration

La figure ci-dessous est un exemple d'écran de mise au point.



Description

Le tableau ci-dessous présente les différentes zones de l'écran de mise au point :

Repère	Element	Fonction
1	Etat des esclaves CANopen	Cette zone affiche tous les esclaves du bus CANopen. Un esclave en défaut est affiché en rouge, lorsque le défaut disparaît, il est affiché en bleu, sinon il est affiché en noir. La sélection d'un esclave met à jour les zones 2, 4 et 5. **Act.** : indique si l'esclave a été activé dans la configuration Sycon (1 activé, 0 désactivé) **Durée de vie (Life T.)** : Life Time.
2	Entrées	Lorsqu'un esclave est sélectionné, cette zone contient la liste des mots qui lui sont associés en entrée. Pour plus de clarté, l'affichage de l'adressage topologique CANopen *(voir page 35)* (%IW\3.1\0.0.0.0) et de la zone mémoire réservée dans l'automate (%MW4) est proposé.

Repère	Element	Fonction
3	Valeur en sortie	Lorsqu'un mot de sortie est sélectionné dans la zone 4, il est possible de modifier sa valeur en saisissant une nouvelle valeur puis en cliquant sur le bouton **OK**.
4	Sorties	Lorsqu'un esclave est sélectionné, cette zone contient la liste des mots qui lui sont associés en sortie. Pour plus de clarté, l'affichage de l'adressage topologique CANopen *(voir page 35)* (%QW\3.1\0.0.0.0) et de la zone mémoire réservée dans l'automate (%MW14) est proposé.
5	Informations sur ...	Lorsqu'un esclave est sélectionné (cliquez dans la zone 1), cette zone contient son dernier message de diagnostic et pour obtenir des informations sur la carte TSX CPP 110 il suffit de cliquer sur l'entête du tableau.
6	Requête à émettre	Cette zone permet d'émettre une requête SDO. La syntaxe des paramètres est identique à celle utilisée pour effectuer des transferts SDO par les requêtes READ_VAR et WRITE_VAR *(voir page 57)*. L'appui sur le bouton **Saisie requête** fait apparaître les zones de saisie le la requête.

3.5 Diagnostics

Objet de cette section

Cette section décrit les outils de diagnostic équipement et logiciel disponibles pour la carte PCMCIA TSX CPP 110.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Diagnostics à l'aide des voyants d'état sur la carte PCMCIA TSX CPP 110	70
Données de diagnostic	71
Comment effectuer un diagnostic	74

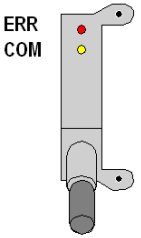
Diagnostics à l'aide des voyants d'état sur la carte PCMCIA TSX CPP 110

Présentation

Les voyants sur la carte vous permettent de voir le statut de la carte et du bus CANopen. En fonctionnement normal, le voyant ERR est éteint et le voyant COM est allumé.

Illustration

La figure ci-dessous représente la position des deux voyants, ERR et COM.



Diagnostics

Selon le statut des voyants, les diagnostics sont les suivants :

Etat des voyants		**Signification**
ERR (rouge)	**COM** (jaune)	
Eteint	Eteint	Carte sans alimentation ou transfert de configuration en cours
	Clignotant (Irrégulier)	Pas de configuration sur la carte
	Clignotant (Régulier)	Carte configurée et prête, bus non actif ou pas de microprogramme CANopen
	Allumé	Bus configuré et actif, pas d'erreur
Allumé en permanence	Eteint	Erreur détectée, automate du bus arrêté
	Clignotant	Carte configurée et prête mais impossibilité de communiquer avec un équipement périphérique distant (p.ex. câble du bus CAN déconnecté), ou tous les équipements périphériques configurés font état d'une erreur. Erreur sur la carte, erreur de configuration ou erreur de synchronisation entre la carte et l'automate (pour plus d'informations, veuillez consulter les données diagnostics de statut du module)
	Allumé	Bus configuré et actif, au moins un des souscripteurs du bus ne peut être atteint ou signale une erreur

Données de diagnostic

Présentation

Durant les diagnostics, les premières données à être utilisées sont les mots à échanges implicites %IW *(voir page 92)*.

Il existe également des données de diagnostic qui peuvent être consultées en écrivant certaines lignes de programmes dans l'automate.

La fonction de communication SEND_REQ permet de réaliser des diagnostics *(voir page 65)*:

- Sur l'esclave de son choix (1 à 127, une requête par esclave),
- Avec la version de la carte PCMCIA (128),
- Dans l'état de la carte (129),

Et est utilisé pour obtenir l'historique des messages d'erreur (130).

NOTE : Le code 129 permet de recevoir les mêmes informations que celles contenues dans les mots d'entrée à échange implicite *(voir page 93)*.

Les informations fournies viennent de la carte PCMCIA et sont mises à jour régulièrement.

Le tableau de réception de requête contient les informations décrites dans les paragraphes suivants.

NOTE : Les informations sont données dans les tables d'octets. Tant que la possibilité de demander toute ou une partie de la table est prise en compte, il est nécessaire de tenir compte des mots de poids fort et des mots de poids faible de la table `%MWi:L`.

NOTE : Les informations de diagnostic sont conformes à la norme CAN. Des informations de référence sont disponibles sur ce site : `http://www.can-cia.de`.

Diagnostics esclaves

Le tableau suivant décrit les informations reçues après une requête de diagnostics sur un esclave (code 1 à 127).

Rang d'octet	Description
0	Bits d'état de l'équipement : ● Bit 0 : Pas de réponse ● Bit 1 : Dépassement de la table de l'historique des messages d'erreur ● Bit 2 : Erreur paramètres ● Bit 3 : Gestion équipement actif ● Bits 4 à 6 : Réservé ● Bit 7 : Désactivé
1 et 2	Informations complémentaires lues au démarrage du bus sur l'objet standard 16#1000 (normalisation CAN)
3 et 4	Numéro de profil lu au démarrage du bus sur l'objet standard 16#1000 (normalisation CAN)
5	Statut esclave : ● 1: Déconnecté ● 2: Connexion en cours ● 3: En préparation ● 4: Prêt ● 5: Fonctionnement ● 127: Prêt ou manquant
6	Code d'erreur *(voir page 106)* (code de la dernière erreur générée par l'esclave)
7	Nombre de blocs d'informations urgentes sur l'esclave (0 à 5). **Remarque** : Ces blocs sont ajoutés à la fin du tableau. Des détails sur les blocs typiques sont disponibles dans le paragraphe suivant.

Description d'un bloc d'informations sur l'esclave :

Le tableau suivant décrit un bloc d'informations typique.

Rang octet	Description
0 et 1	Code d'erreur
2	Valeur du registre d'erreur, objet 16#1001 de l'esclave (normalisation CAN)
3 à 6	Valeur du registre du statut fabricant spécifique, objet 16#1002 (normalisation CAN)
7	Réservé

Diagnostics sur une version de carte

Le tableau suivant décrit les informations reçues après une requête de diagnostics sur une carte PCMCIA (code 128).

Rang octet	Description
0 à 7	Version du microprogramme (chaîne de caractères)
8 à 10	Date de création de la version du microprogramme (codée BCD, JJ.MM.AA)
11 à 13	Date de fabrication (codée BCD, JJ.MM.AA)
14 à 17	Numéro de série (codé BCD)
18 à 25	Nom de protocole (ASCII, sans fin de chaîne de caractères, p.ex. : "CANopen")

Historique des messages d'erreur

Le tableau suivant décrit les informations reçues après une requête de rapport historique (code 130).

Rang octet	Description
0 et 1	Nombre de blocs d'erreur **Remarque** : Ces blocs sont ajoutés à la fin du tableau. Des détails sur les blocs typiques sont disponibles dans le paragraphe suivant.
2 à 49	Contenus du bloc d'erreur (maximum 8 blocs)

Description d'un bloc d'informations du rapport historique

Le tableau suivant décrit un bloc d'informations typique.

Rang octet	Description
0	Code de service
1	ID concernée
2 et 3	Code des messages d'erreur. *(voir page 102)*
4 et 5	Détail du code d'erreur *(voir page 104)*

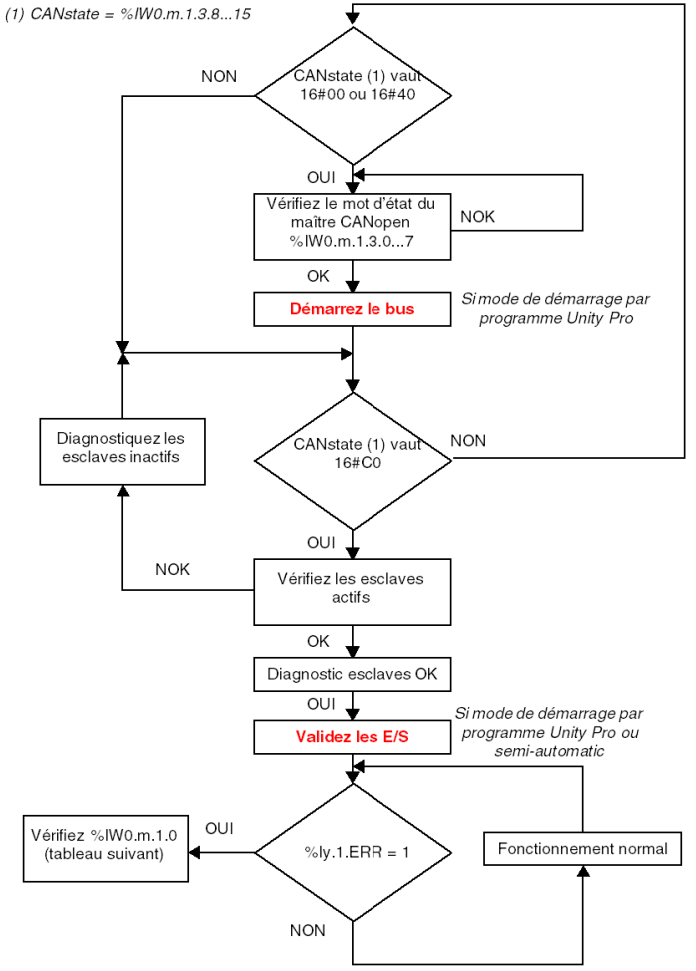
Comment effectuer un diagnostic

Présentation

La recherche d'un défaut sur le bus CANopen s'effectue tout d'abord à l'aide des voyants de la carte PCMCIA. Ensuite, vous pouvez utiliser la procédure décrite ci-après qui décrit la gestion du démarrage du bus ainsi que les vérifications à effectuer en utilisant les objets langage *(voir page 77)* fournis par l'automate.

Marche à suivre

Le schéma suivant indique les différentes phases de la marche à suivre.



Comment vérifier %IW0.m.1.0

Ce tableau décrit les actions à effectuer pour obtenir un diagnostic précis à l'aide des bits x8 à x15 de %IW0.m.1.0.

Si ...	Alors ...
bit 8 = 1	c'est une erreur de configuration. Vérifiez le détail des codes d'erreur *(voir page 96)* dans les mots : ● %IW0.m.1.1 ● %IW0.m.1.2
bit 9 = 1	c'est une erreur de transfert de PDO. Contactez le support technique Schneider.
bit 10 = 1	c'est une erreur de transfert de SDO Vérifiez le détail des codes d'erreur *(voir page 96)* dans les mots : ● %IW0.m.1.1 ● %IW0.m.1.2 Vérifiez l'historique des erreurs de messagerie *(voir page 105)*.

bit 11 = 1	c'est un défaut de la carte PCMCIA.
	● Vérifiez le détail des codes d'erreur dans %IW0.m.1.1.
	● Vérifiez le contenu de %IW0.m.1.3 :
	  ● bit 0 : erreur de paramètres, la source du défaut est indiquée dans %IW0.m.1.4
	  ● bit 1 : les sorties sont à zéro suite à la défaillance d'un esclave (Autoclear ON), la source du défaut est indiquée dans %IW0.m.1.4
	  ● bit 3 : défaut grave, la carte n'est pas active sur le bus
	  ● bit 7 : connexion défectueuse entre la carte et le boîtier de raccordement.

bit 12 = 1	c'est un défaut du bus (bus non démarré ou défaut de transmission détecté). ● Vérifiez le compteur du nombre d'erreurs du bus %IW0.m.1.5, s'il est différent de zéro, vérifiez la ligne ● Vérifiez le compteur du nombre d'arrêts du bus, s'il s'incrémente, vérifiez la ligne et redémarrez le bus. **Remarque** : en mode de démarrage non automatique, le bit de démarrage du bus est %QW0.m.1.0.
bit 13 = 1	c'est un défaut sur un esclave : défaut de communication ou E/S non validées. ● Déterminez la dernière source d'erreur contenue dans %IW0.m.1.4 ● Déterminez tous les esclaves actifs sur le bus en consultant les mots d'état du bus %IW0.m.1.16 à %IW0.m.1.23 ● Effectuez un diagnostic des esclaves en défaut à l'aide d'une requête de diagnostic *(voir page 65)*. **Remarque** : en mode de démarrage non automatique, le bit de démarrage des E/S est %QW0.m.1.1.

Si ...	Alors ...
bit 14 = 1	c'est un défaut sortie : les sorties sont positionnées en conditions de repli. ● Vérifiez que l'automate est en RUN ● Vérifiez que la tâche associée au module est active ● Testez le bit %IW0.m.1.0.12 (Défaut bus) et le bit %IW0.m.1.0.13 (Défaut esclave). **Remarque** : en mode de démarrage non automatique, testez les bits %QW0.m.1.0 et %QW0.m.1.1.
bit 15 = 1	un nouveau diagnostic est disponible pour un ou plusieurs esclaves. ● Déterminez les esclaves concernés grâce aux mots d'état %IW0.m.1.16 à %IW0.m.1.23 ● Effectuez un diagnostic *(voir page 65)* du ou des esclaves concernés

Objets langage CANopen

4

Objet de ce chapitre

Ce chapitre décrit les objets langage associés à la voie de communication CANopen.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
4.1	Objets langage et IODDT pour la communication CANopen	78
4.2	Objets langage et IODDT générique applicables aux protocoles de communication	87
4.3	Objets langage du IODDT spécifique à CANopen	91
4.4	IODDT de type T_GEN_MOD applicable à tous les modules	98
4.5	Objets langage de configuration CANopen	100
4.6	Codes d'erreur CANopen	102

4.1 Objets langage et IODDT pour la communication CANopen

Objectif de cette section

Cette section présente les généralités des principaux points des objets langage et des IODDT pour la communication CANopen.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Présentation des objets langage pour la communication CANopen	79
Objets langage à échange implicite associés à la fonction métier	80
Objets langage à échange explicite associés à la fonction métier	81
Gestion des échanges et comptes rendus avec des objets explicites	83

Présentation des objets langage pour la communication CANopen

Général

Les IODDT sont prédéfinis par le constructeur. Ils contiennent des objets langage d'entrées/sorties appartenant à une voie d'un module spécifique.

La communication CANopen a deux IODDT associés :

- `T_COM_STS_GEN` applicable à tous les protocoles de communication, sauf Fipio et Ethernet ;
- `T_COM_CPP110` propre à la communication CANopen.

NOTE : les variables IODDT peuvent être créées de deux façons :

- à l'aide de l'onglet Objets d'E/S *(voir Unity Pro, Modes de marche)*,
- à l'aide de l'Editeur de données. *(voir Unity Pro, Modes de marche)*

Types d'objets langage

Dans chacun des IODDT se trouve un ensemble d'objets langage permettant de les commander et de vérifier leur fonctionnement.

Il existe deux types d'objets langage :

- **les objets à échange implicite**, qui sont échangés automatiquement à chaque cycle de la tâche associée au module ;
- **les objets à échange explicite**, qui sont échangés à la demande de l'application, en utilisant des instructions d'échange explicite.

Les échanges implicites concernent l'état des modules, les signaux de communication, les esclaves, etc.

Les échanges explicites permettent de paramétrer le module et de le diagnostiquer.

Objets langage à échange implicite associés à la fonction métier

Présentation

Une interface métier intégrée ou l'ajout d'un module enrichit automatiquement le projet d'objets langage permettant de programmer cette interface ou ce module.

Ces objets correspondent aux images des entrées/sorties et informations logicielles du module ou de l'interface métier intégrée.

Rappels

Les entrées (`%I` et `%IW`) du module sont mises à jour dans la mémoire automate en début de tâche, alors que l'automate est en mode RUN ou STOP.

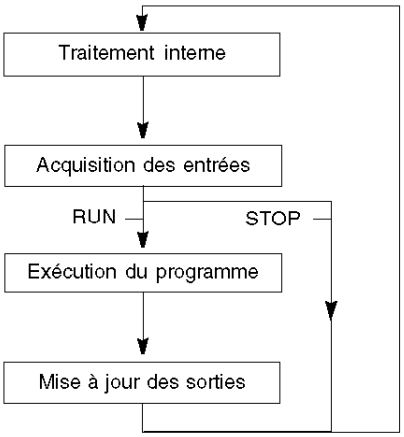
Les sorties (`%Q` et `%QW`) sont mises à jour en fin de tâche, uniquement lorsque l'automate est en mode RUN.

NOTE : Lorsque la tâche est en mode STOP, suivant la configuration choisie :

- les sorties sont mises en position de repli (mode repli)
- les sorties sont maintenues à leur dernière valeur (mode maintien)

Illustration

Le schéma ci-dessous illustre le cycle de fonctionnement relatif à une tâche automate (exécution cyclique).



Objets langage à échange explicite associés à la fonction métier

Introduction

Les échanges explicites sont des échanges réalisés à la demande de l'utilisateur du programme, et à l'aide des instructions suivantes :

- READ_STS *(voir Unity Pro, Gestion des E/S, Bibliothèque de blocs)* (lecture des mots d'état)
- WRITE_CMD *(voir Unity Pro, Gestion des E/S, Bibliothèque de blocs)* (écriture des mots de commande)
- WRITE_PARAM *(voir Unity Pro, Gestion des E/S, Bibliothèque de blocs)* (écriture des paramètres d'ajustement)
- READ_PARAM *(voir Unity Pro, Gestion des E/S, Bibliothèque de blocs)* (lecture des paramètres d'ajustement)
- SAVE_PARAM *(voir Unity Pro, Gestion des E/S, Bibliothèque de blocs)* (enregistrement des paramètres d'ajustement)
- RESTORE_PARAM *(voir Unity Pro, Gestion des E/S, Bibliothèque de blocs)* (restauration des paramètres d'ajustement)

Ces échanges s'appliquent à un ensemble d'objets %MW de même type (état, commandes ou paramètres) appartenant à une voie.

Ces objets peuvent :

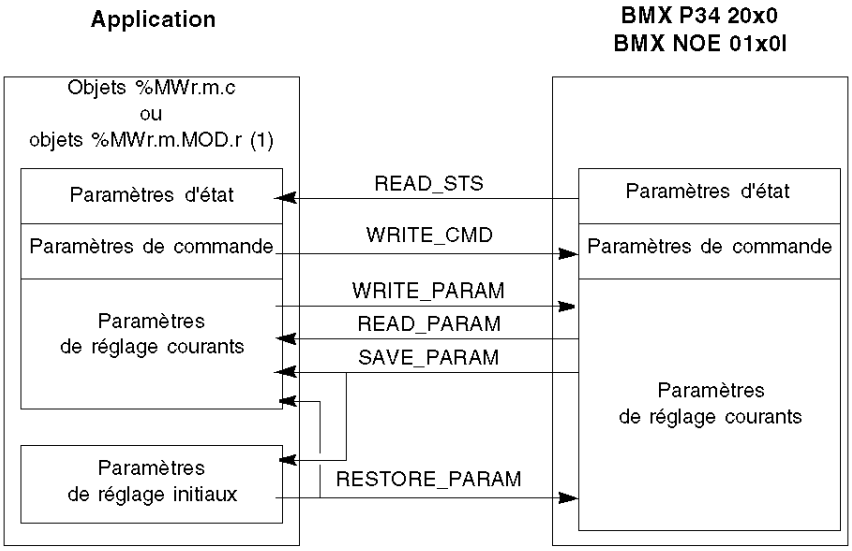
- fournir des informations sur le module (par exemple, le type d'erreur détectée dans une voie),
- commander le module (grâce à un commutateur, par exemple),
- définir les modes de fonctionnement du module (enregistrement et restauration des paramètres d'ajustement pendant l'exécution de l'application).

NOTE : afin d'éviter plusieurs échanges explicites simultanés pour la même voie, il est nécessaire de tester la valeur du mot EXCH_STS (`%MWr.m.c.0`) de l'IODDT associé à la voie avant d'appeler une fonction élémentaire (EF) utilisant cette voie.

NOTE : Les échanges explicites ne sont pas pris en charge lorsque des modules d'E/S numériques et analogiques Modicon M340 sont configurés derrière un module adaptateur d'E/S distantes Ethernet M340 dans une configuration d'E/S Ethernet Quantum. Il n'est donc pas possible de configurer les paramètres d'un module à partir de l'application de l'automate en cours de fonctionnement.

Principe général d'utilisation des instructions explicites

Le schéma ci-après présente les différents types d'échanges explicites possibles entre l'application et le module.



(1) Seulement avec les instructions READ_STS et WRITE_CMD.

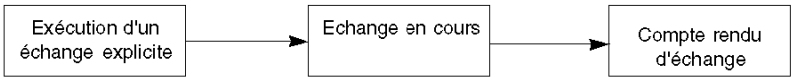
Gestion des échanges

Pendant un échange explicite, vérifiez les performances pour vérifier que les données ne sont prises en compte que lorsque l'échange a été correctement exécuté.

Pour cela, deux types d'information sont disponibles :

- les informations relatives à l'échange en cours, *(voir page 85)*
- le rapport d'échange. *(voir page 86)*

Le diagramme ci-après décrit le principe de gestion d'un échange.



NOTE : afin d'éviter plusieurs échanges explicites simultanés pour la même voie, il est nécessaire de tester la valeur du mot EXCH_STS (`%MWr.m.c.0`) de l'IODDT associé à la voie avant d'appeler une fonction élémentaire (EF) utilisant cette voie.

Gestion des échanges et comptes rendus avec des objets explicites

Vue d'ensemble

Lorsque les données sont échangées entre la mémoire automate et le module, la prise en compte par le coupleur peut nécessiter plusieurs cycles de la tâche. Pour gérer les échanges, tous les IODDT possèdent deux mots :

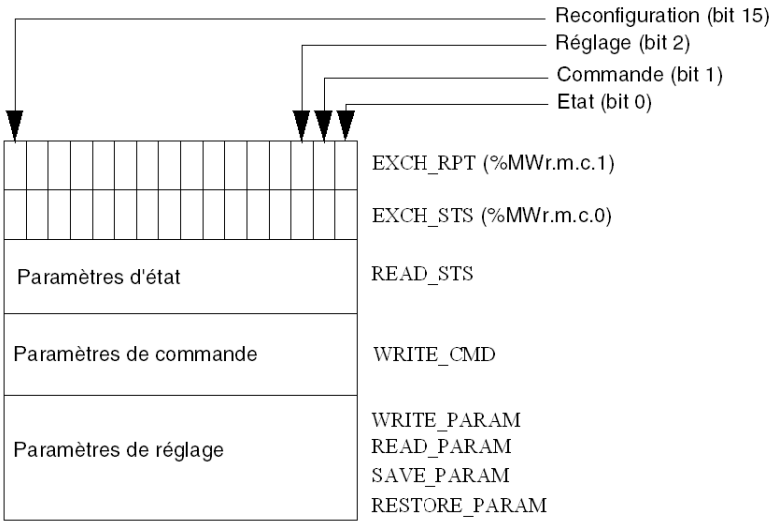
- `EXCH_STS` (`%MWr.m.c.0`) : échange en cours
- `EXCH_RPT` (`%MWr.m.c.1`) : compte rendu

NOTE : selon l'emplacement du module, la gestion des échanges explicites (`%MW0.0.MOD.0.0`, par exemple) ne sera pas détectée par l'application :

- Pour les modules en rack, les échanges explicites ont lieu immédiatement sur le bus automate local et se terminent avant la fin de la tâche d'exécution, afin que le `READ_STS`, par exemple, soit toujours terminé quand le bit `%MW0.0.mod.0.0` est vérifié par l'application.
- Pour le bus distant (Fipio par exemple), les échanges explicites ne sont pas synchronisés avec la tâche d'exécution, afin que la détection par l'application soit possible.

Illustration

L'illustration ci-dessous présente les différents bits significatifs pour la gestion des échanges :



Description des bits significatifs

Chacun des bits des mots EXCH_STS (%MWr.m.c.0) et EXCH_RPT (%MWr.m.c.1) est associé à un type de paramètre :

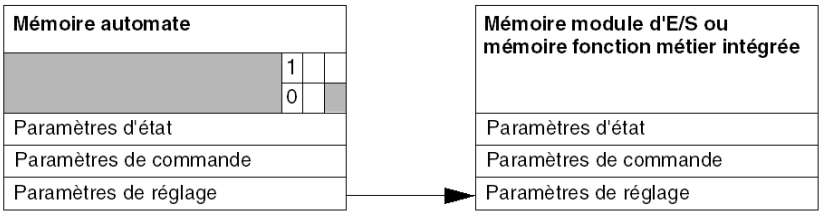
- Les bits de rang 0 sont associés aux paramètres d'état :
  - le bit STS_IN_PROGR (%MWr.m.c.0.0) indique si une demande de lecture des mots d'état est en cours ;
  - le bit STS_ERR (%MWr.m.c.1.0) précise si une demande de lecture des mots d'état est acceptée par la voie du module.
- Les bits de rang 1 sont associés aux paramètres de commande :
  - le bit CMD_IN_PROGR (%MWr.m.c.0.1) indique si des paramètres de commande sont envoyés à la voie du module ;
  - le bit CMD_ERR (%MWr.m.c.1.1) précise si les paramètres de commande sont acceptés par la voie du module.
- Les bits de rang 2 sont associés aux paramètres de réglage :
  - le bit ADJ_IN_PROGR (%MWr.m.c.0.2) indique si des paramètres de réglage sont échangés avec la voie du module (par WRITE_PARAM, READ_PARAM, SAVE_PARAM, RESTORE_PARAM) ;
  - le bit ADJ_ERR (%MWr.m.c.1.2) précise si les paramètres de réglage sont acceptés par le module. Si l'échange s'est correctement déroulé, le bit passe à 0.
- Les bits de rang 15 indiquent une reconfiguration sur la voie **c** du module depuis la console (modification des paramètres de configuration et démarrage à froid de la voie).
- Les bits *r*, *m* et *c* représentent les éléments suivants :
  - le bit **r** représente le numéro du rack ;
  - le bit **m** représente la position du module dans le rack ;
  - le bit **c** représente le numéro de voie dans le module.

NOTE : r représente le numéro du rack, **m** représente la position du module dans le rack, **c** représente le numéro de voie dans le module.

NOTE : Les mots d'échange et de compte rendu existent aussi au niveau des modules EXCH_STS (%MWr.m.MOD) et EXCH_RPT (%MWr.m.MOD.1) selon le type IODDT T_GEN_MOD.

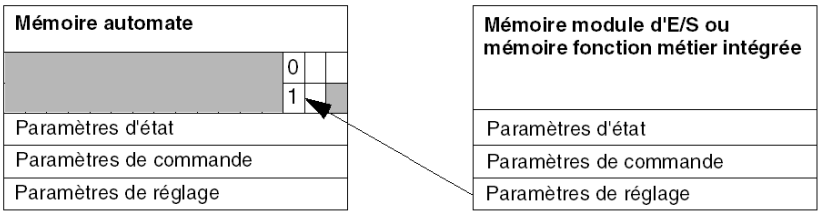
Exemple

Phase 1 : émission de données à l'aide de l'instruction `WRITE_PARAM`.



Lorsque l'instruction est scrutée par le processeur automate, le bit **Echange en cours** est réglé sur 1 dans `%MWr.m.c`.

Phase 2 : analyse des données par le module d'E/S et compte rendu.



Lorsque les données sont échangées entre la mémoire automate et le module, le bit `ADJ_ERR` (`%MWr.m.c.1.2`) gère l'acquittement par le module.

Ce bit génère les comptes rendus suivants :
- **0 :** échange correct,
- **1 :** échange défectueux).

NOTE : il n'existe pas de paramètre de réglage au niveau du module.

Indicateurs d'exécution d'échange explicite : EXCH_STS

Le tableau ci-dessous présente les bits de contrôle des échanges explicites : `EXCH_STS` (`%MWr.m.c.0`).

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	R	Lecture des mots d'état de la voie en cours	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Echange de paramètres de commande en cours	%MWr.m.c.0.1

Symbole standard	Type	Accès	Signification	Adresse
ADJ_IN_PROGR	BOOL	R	Echange de paramètres de réglage en cours	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	Reconfiguration du module en cours	%MWr.m.c.0.15

NOTE : Si le module n'est pas présent ou est déconnecté, les objets à échange explicite (`READ_STS`, par exemple) ne sont pas envoyés au module (`STS_IN_PROG` (%MWr.m.c.0.0) = 0), mais les mots sont rafraîchis.

Compte rendu d'échange explicite : EXCH_RPT

Le tableau ci-dessous présente les bits de compte rendu : `EXCH_RPT` (`%MWr.m.c.1`).

Symbole standard	Type	Accès	Signification	Adresse
STS_ERR	BOOL	R	Erreur de lecture des mots d'état de la voie (1 = échec)	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Erreur lors d'un échange de paramètres de commande (1 = échec)	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Erreur lors de l'échange de paramètres de réglage (1 = échec)	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Erreur lors de la reconfiguration de la voie (1 = échec)	%MWr.m.c.1.15

Utilisation du module de comptage

Le tableau suivant décrit les étapes réalisées entre un module de comptage et le système après une mise sous tension.

Etape	Action
1	Sous tension
2	Le système envoie les paramètres de configuration.
3	Le système envoie les paramètres de réglage à l'aide de la méthode WRITE_PARAM. **Remarque :** une fois l'opération terminée, le bit %MWr.m.c.0.2 passe à 0.

Si vous utilisez une commande WRITE_PARAM au début de l'application, vous devez attendre que le bit %MWr.m.c.0.2 passe à 0.

4.2 Objets langage et IODDT générique applicables aux protocoles de communication

A propos de cette section

Cette section présente les objets langage et l'IODDT générique applicables à tous les protocoles de communication, sauf Fipio et Ethernet.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Détails des objets à échange implicite de type d'IODDT T_COM_STS_GEN	88
Détails des objets à échange explicite de l'IODDT de type T_COM_STS_GEN	89

Détails des objets à échange implicite de type d'IODDT T_COM_STS_GEN

Introduction

Le tableau ci-dessous présente les objets à échange implicite de l'IODDT de type T_COM_STS_GEN, qui s'appliquent à tous les protocoles de communication sauf Fipio et Ethernet.

Bit d'erreur

Le tableau ci-dessous présente la signification du bit d'erreur détectée CH_ERROR (%Ir.m.c.ERR).

Symbole standard	Type	Accès	Signification	Adresse
CH_ERROR	EBOOL	R	Bit erreur de la voie de communication.	%Ir.m.c.ERR

Détails des objets à échange explicite de l'IODDT de type T_COM_STS_GEN

Présentation

Cette partie présente les objets à échange explicite de l'IODDT de type T_COM_STS_GEN qui s'appliquent à tous les protocoles de communication sauf Fipio. Elle regroupe les objets de type mot, dont les bits ont une signification particulière. Ces objets sont présentés en détail ci-dessous.

Exemple de déclaration d'une variable : IODDT_VAR1 de type T_COM_STS_GEN.

Remarques

- De manière générale, la signification des bits est donnée pour l'état 1 de ce bit. Dans les cas spécifiques, chaque état du bit est expliqué.
- Tous les bits ne sont pas utilisés.

Indicateurs d'exécution d'un échange explicite : EXCH_STS

Le tableau ci-dessous présente la signification des bits de contrôle d'échange de la voie EXCH_STS (%MWr.m.c.0).

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	L	Lecture des mots d'état de la voie en cours.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	L	Echange de paramètres actuels en cours.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	L	Echange de paramètres de réglage en cours.	%MWr.m.c.0.2

Compte rendu d'échanges explicites : EXCH_RPT

Le tableau ci-dessous présente la signification des bits de compte rendu EXCH_RPT (%MWr.m.c.1).

Symbole standard	Type	Accès	Signification	Adresse
STS_ERR	BOOL	L	Défaut de lecture des mots d'état de la voie.	%MWr.m.c.1.0
CMD_ERR	BOOL	L	Défaut lors d'un échange de paramètres de commande.	%MWr.m.c.1.1
ADJ_ERR	BOOL	L	Défaut lors d'un échange de paramètres de réglage.	%MWr.m.c.1.2

Défauts de voie standard, CH_FLT

Le tableau ci-dessous présente la signification des bits du mot d'état CH_FLT (%MWr.m.c.2). La lecture est effectuée par un READ_STS (IODDT_VAR1).

Symbole standard	Type	Accès	Signification	Adresse
NO_DEVICE	BOOL	L	Aucun équipement ne fonctionne sur la voie.	%MWr.m.c.2.0
1_DEVICE_FLT	BOOL	L	Un équipement sur la voie est en défaut.	%MWr.m.c.2.1
BLK	BOOL	L	Défaut bornier (non connecté).	%MWr.m.c.2.2
TO_ERR	BOOL	L	Erreur de Time out (câblage défectueux).	%MWr.m.c.2.3
INTERNAL_FLT	BOOL	L	Défaut interne ou autotest de la voie.	%MWr.m.c.2.4
CONF_FLT	BOOL	L	Configurations matérielle et logicielle différentes.	%MWr.m.c.2.5
COM_FLT	BOOL	L	Défaut de communication avec l'automate.	%MWr.m.c.2.6
APPLI_FLT	BOOL	L	Défaut applicatif (défaut de réglage ou de configuration).	%MWr.m.c.2.7

4.3 Objets langage du IODDT spécifique à CANopen

Objet de cette section

Cette section décrit les objets langage implicites et explicites du IODDT spécifique à CANopen, T_COM_CPP110.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Détails des objets à échange implicite de type T_COM_CPP110 de l'IODDT	92
Détails des objets à échange implicite de type T_COM_CPP110 n'appartenant pas à l'IODDT	96
Objets langage à échange explicite de l'IODDT T_COM_CPP110	97

Détails des objets à échange implicite de type T_COM_CPP110 de l'IODDT

Présentation

Les tableaux ci-dessous décrivent tout les objets langage à échange implicite de l'IODDT de type `T_COM_CPP110` pour une communication CANopen avec la carte PCMCIA TSX CPP 110.

Bit d'erreur

Le tableau ci-dessous répertorie les différents objets bit pour échange implicite.

Symbole standard	Type	Accès	Signification	Adresse
CH_ERROR	EBOOL	R	Bit d'erreur de la voie de communication.	%I0.m.0.ERR

Bits d'état de la communication

Le tableau ci-dessous présente le mot d'erreur pour la gestion du bus CANopen. Il est détaillé bit par bit.

Symbole standard	Type	Accès	Signification	Adresse
CAN_FLT	BOOL	R	Erreur de voie, logique **OU** entre tous les bits qui suivent excepté le bit 15.	%IW0.m.1.0.0
SOFT_CONF	BOOL	R	Configuration incorrecte	%IW0.m.1.0.8
PDO_EXCH_FLT	BOOL	R	Erreur dans l'échange E/S (PDO)	%IW0.m.1.0.9
MSG_EXCH_FLT	BOOL	R	Erreur d'échange messagerie (SDO)	%IW0.m.1.0.10
CARD_FLT	BOOL	R	Erreur de carte (carte manquante ou pas prête)	%IW0.m.1.0.11
BUS_FLT	BOOL	R	Défaillance de bus (au moins un événement de type **erreur de bus** a été généré)	%IW0.m.1.0.12
SLAVE_FLT_B	BOOL	R	Défaillance de l'esclave (une erreur de communication a été détectée sur au moins un esclave)	%IW0.m.1.0.13
OUTP_FLT	BOOL	R	Erreur de sortie (les sorties sont positionnées en conditions de repli)	%IW0.m.1.0.14
SLAVE_DIAG	BOOL	R	Nouveaux diagnostics esclaves disponibles (un nouveau diagnostic est disponible pour au moins un esclave)	%IW0.m.1.0.15

Bits d'état du maître CANopen

Le tableau ci-dessous présente le mot d'erreur du bus maître CANopen `MAST_STS` (`%IW0.m.1.3`). Il est détaillé bit par bit.

Symbole standard	Type	Accès	Signification	Adresse
PARAM_FLT	BOOL	R	Erreur paramètres	%IW0.m.1.3.0
SLAVE_OUTP_FLT	BOOL	R	Indique que les sorties sont à zéro après la défaillance d'un esclave, Autoclear ON.	%IW0.m.1.3.1
NO_BUS_EXCH	BOOL	R	Pas d'échange sur le bus (aucun esclave ne communique)	%IW0.m.1.3.2
CARD_NO_ACT	BOOL	R	Erreur grave, la carte n'est pas active sur le bus	%IW0.m.1.3.3
FEW_BUS_ERR	BOOL	R	Au moins un événement d'**erreur de bus** a été détecté	%IW0.m.1.3.4
CARD_ACC_PROH	BOOL	R	Le processeur n'a pas encore été autorisé à accéder à la carte.	%IW0.m.1.3.5
TIMEOUT_MSG	BOOL	R	Temporisation dépassée lors de l'envoi de messages CAN	%IW0.m.1.3.6
CNX_FLT	BOOL	R	Connexion défectueuse entre la carte et l'unité de connexion.	%IW0.m.1.3.7
-	-	R	Les bits 8 à 15 du `MAST_STS` constituent un octet dont la valeur indique le mode de fonctionnement. • 16#00: mode hors ligne • 16#40: bus en mode STOP • 16#80: les sorties sont en mode sécurité (à zéro) • 16#C0: bus en mode RUN	%IW0.m.1.3.8 à %IW0.m.1.3.15

Mot d'état pour les équipements de bus

Le tableau ci-dessous présente les mots d'état pour les équipements du bus CANopen :

Symbole standard	Type	Accès	Signification	Adresse
SLAVE_STS	INT	R	L'octet de poids faible (bits 0 à 7) contient l'adresse de l'esclave qui a généré la dernière erreur. L'octet de poids fort (bits x8 à x15) contient le dernier code d'erreur.	%IW0.m.1.4

Mots d'état du bus

Le tableau ci-dessous présente les mots d'état pour le bus CANopen. Ces mots sont divisés en bits qui représentent chacun un esclave du bus.

Symbole standard	Type	Accès	Signification	Adresse
SLAVE_ACTIV_0	BOOL	R	Esclave 0 actif sur le bus.	%IW0.m.1.8.0
SLAVE_ACTIV_1	BOOL	R	Esclave 1 actif sur le bus.	%IW0.m.1.8.1
...	...	...	...	...
SLAVE_ACTIV_15	BOOL	R	Esclave 15 actif sur le bus.	%IW0.m.1.8.15
SLAVE_ACTIV_16	BOOL	R	Esclave 16 actif sur le bus.	%IW0.m.1.9.0
...	...	...	...	...
SLAVE_ACTIV_31	BOOL	R	Esclave 31 actif sur le bus.	%IW0.m.1.9.15
SLAVE_ACTIV_32	BOOL	R	Esclave 32 actif sur le bus.	%IW0.m.1.10.0
...	...	...	...	...
SLAVE_ACTIV_47	BOOL	R	Esclave 47 actif sur le bus.	%IW0.m.1.10.15
SLAVE_ACTIV_48	BOOL	R	Esclave 48 actif sur le bus.	%IW0.m.1.11.0
...	...	...	...	...
SLAVE_ACTIV_63	BOOL	R	Esclave 63 actif sur le bus.	%IW0.m.1.11.15
SLAVE_ACTIV_64	BOOL	R	Esclave 64 actif sur le bus.	%IW0.m.1.12.0
...	...	...	...	...
SLAVE_ACTIV_79	BOOL	R	Esclave 79 actif sur le bus.	%IW0.m.1.12.15
SLAVE_ACTIV_80	BOOL	R	Esclave 80 actif sur le bus.	%IW0.m.1.13.0
...	...	...	...	...
SLAVE_ACTIV_95	BOOL	R	Esclave 95 actif sur le bus.	%IW0.m.1.13.15
SLAVE_ACTIV_96	BOOL	R	Esclave 96 actif sur le bus.	%IW0.m.1.14.0
...	...	...	...	...
SLAVE_ACTIV_111	BOOL	R	Esclave 111 actif sur le bus.	%IW0.m.1.14.15
SLAVE_ACTIV_112	BOOL	R	Esclave 112 actif sur le bus.	%IW0.m.1.15.0
...	...	...	...	...
SLAVE_ACTIV_127	BOOL	R	Esclave 127 actif sur le bus.	%IW0.m.1.15.15

Mot de sortie

Le tableau ci-dessous présente les mots de sortie `%QW0.m.1.0` de la carte CANopen PCMCIA. Il est détaillé bit par bit.

Symbole standard	Type	Accès	Signification	Adresse
ACT_BUS_CONF	BOOL	RW	Ce bit est uniquement utilisé lorsque le démarrage du bus est géré par l'application : • 1: active la configuration du bus • 0: désactive la configuration du bus	%QW0.m.1.0.0
ACT_DATA_TR	BOOL	RW	Ce bit est utilisé lorsque le démarrage est semi-automatique ou géré par l'application • 1: active le transfert de données sur le bus. • 0: désactive le transfert de données sur le bus.	%QW0.m.1.0.1
INIT_ERR_BIT	BOOL	RW	Initialise les bits d'erreur : • Erreurs d'E/S, • erreurs de messagerie, • erreurs d'historique.	%QW0.m.1.0.2
INIT_CARD	BOOL	RW	Initialise la carte PCMCIA. Ce bit démarre à chaud la carte et est uniquement utilisé lorsque le démarrage est géré par l'application.	%QW0.m.1.0.3

NOTE : Les bits de commande 2 et 3 ne sont pas automatiquement remis à zéro par l'application.

Détails des objets à échange implicite de type T_COM_CPP110 n'appartenant pas à l'IODDT

Présentation

Le tableau suivant décrit les objets langage CANopen qui n'appartiennent pas à un IODDT spécifique mais qui peuvent être directement utilisés sur la base de leur nombre.

Objets mot d'entrée

Le tableau ci-dessous répertorie les différents objets mot d'entrée pour échange implicite qui n'appartiennent pas à un IODDT.

Objet (1)	Fonction	Signification
%IW0.m.1.1	Mot d'erreur	Ce mot contient un code d'erreur module *(voir page 102)* (erreur dernière configuration ou E/S)
%IW0.m.1.2	Mot d'erreur	Ce mot contient un code d'erreur module détaillé *(voir page 104)* (erreur dernière configuration ou E/S)
%IW0.m.1.5	Mot d'état du compteur	Compteur du nombre d'erreurs de bus
%IW0.m.1.6	Mot d'état du compteur	Compteur du nombre d'arrêts du bus
%IWy.1.7	Mot d'état du compteur	Compteur du nombre de temporisations sur les messages CAN
%IW0.m.1.16 à %IW0.m.1.23	Mots d'état du bus	Diagnostics disponibles sur le bus, chaque bit réglé sur 1 correspond à un équipement pour lequel des diagnostics sont disponibles (8 mots de 16 bits, soit 128 bits, le maître et 127 esclaves)

Objets langage à échange explicite de l'IODDT T_COM_CPP110

Présentation

Le tableau ci-dessous décrit tous les objets langage à échange explicite de l'IODDT de type T_COM_CPP110 pour une communication CANopen avec la carte PCMCIA TSX CPP 110.

Exemple de déclaration d'une variable : IODDT_VAR1 de type T_COM_CPP110

Mots de gestion des échanges

Les tableaux suivants présentent deux bits explicites de gestion des échanges des variables de type T_COM_CPP110.

Symbole standard	Type	Accès	Signification	Adresse
STS_IN_PROGR	BOOL	R	Echange explicite en cours	%MW0.m.1.1.0
STS_ERR	BOOL	R	Erreur lors de l'échange explicite précédent	%MW0.m.1.1.1

Mot d'état de la carte TSX CPP 110

Le tableau ci-dessous présente le mot d'état de la carte TSX CPP 110. Il est détaillé bit par bit. La lecture est effectuée par une commande READ_STS (IODDT_VAR1).

Symbole standard	Type	Accès	Signification	Adresse
COM_FLT	BOOL	R	Le bus présente une défaillance ou n'est pas initialisé (en mode démarrage géré par l'application)	%MW0.m.1.2.0
FEW_SLAVE_FLT	BOOL	R	L'esclave présente une erreur, un ou plusieurs esclaves comportent des erreurs ou ne sont pas en mode RUN	%MW0.m.1.2.1
CABL_FLT	BOOL	R	L'unité de connexion présente une erreur ou le câblage est défectueux	%MW0.m.1.2.2
CARD_MISS	BOOL	R	La carte PCMCIA : ● n'est pas dans son emplacement, ● n'est pas prête ou ● présente une erreur grave	%MW0.m.1.2.3
CARD_NO_ACC	BOOL	R	La carte PCMCIA : ● est en cours d'initialisation et donc n'est pas prête, ● présente une erreur ou ● n'est pas accessible	%MW0.m.1.2.4
CARD_NO_REC	BOOL	R	La carte ou le type de protocole n'est pas reconnu	%MW0.m.1.2.5
IO_EXCH_FLT	BOOL	R	Erreur dans les échanges d'E/S	%MW0.m.1.2.6
CONF_FLT	BOOL	R	Erreur de configuration ou de paramétrage	%MW0.m.1.2.7

4.4 IODDT de type T_GEN_MOD applicable à tous les modules

Présentation des objets langage de l'IODDT de type T_GEN_MOD

Présentation

Tous les modules des automates Premium ont un IODDT de type `T_GEN_MOD` associé.

Remarques

- De manière générale, la signification des bits est donnée pour l'état 1 de ce bit. Dans les cas spécifiques, chaque état du bit est expliqué.
- Tous les bits ne sont pas utilisés.

Liste des objets

Le tableau ci-dessous présente les différents objets de l'IODDT :

Symbole standard	Type	Accès	Signification	Adresse
MOD_ERROR	BOOL	R	Bit d'erreur de module	%Ir.m.MOD.ERR
EXCH_STS	INT	R	Mot de contrôle d'échange du module.	%MWr.m.MOD.0
STS_IN_PROGR	BOOL	R	Lecture des mots d'état du module en cours.	%MWr.m.MOD.0.0
EXCH_RPT	INT	R	Mot de compte rendu de l'échange.	%MWr.m.MOD.1
STS_ERR	BOOL	R	Erreur lors de la lecture des mots d'état du module.	%MWr.m.MOD.1.0
MOD_FLT	INT	R	Mot d'erreur interne du module.	%MWr.m.MOD.2
MOD_FAIL	BOOL	R	Erreur interne, module en panne.	%MWr.m.MOD.2.0
CH_FLT	BOOL	R	Voie(s) en défaut.	%MWr.m.MOD.2.1
BLK	BOOL	R	Défaut du bornier.	%MWr.m.MOD.2.2
CONF_FLT	BOOL	R	Défaut de configuration matérielle ou logicielle.	%MWr.m.MOD.2.5
NO_MOD	BOOL	R	Module absent ou hors fonctionnement.	%MWr.m.MOD.2.6
EXT_MOD_FLT	BOOL	R	Mot d'erreur interne du module (Extension Fipio uniquement).	%MWr.m.MOD.2.7
MOD_FAIL_EXT	BOOL	R	Défaut interne, module hors service (Extension Fipio uniquement).	%MWr.m.MOD.2.8

Symbole standard	Type	Accès	Signification	Adresse
CH_FLT_EXT	BOOL	R	Voie(s) en défaut (extension Fipio uniquement).	%MWr.m.MOD.2.9
BLK_EXT	BOOL	R	Défaut du bornier (extension Fipio uniquement).	%MWr.m.MOD.2.10
CONF_FLT_EXT	BOOL	R	Défaut de configuration matérielle ou logicielle (extension Fipio uniquement).	%MWr.m.MOD.2.13
NO_MOD_EXT	BOOL	R	Module absent ou hors fonctionnement (Extension Fipio uniquement).	%MWr.m.MOD.2.14

4.5 Objets langage de configuration CANopen

Objets langage associés à la configuration

Présentation

Cette page décrit tous les objets langage de configuration pour la communication CANopen avec la carte PCMCIA TSX CPP 110 qui peuvent être affichés par le programme d'application.

Constantes internes

Le tableau suivant décrit les constantes internes :

Objet	Type	Accès	Signification
%KW0.m.1.0	INT	R	Valeur constante utilisée par le système Octet de poids faible : 16#00 Octet de poids fort : 16#37
%KW0.m.1.1	INT	R	Bits de configuration ● Mode repli de sortie lorsque l'automate passe en mode STOP : Bit 0 = 0 : RAZ Bit 0 = 1 : Maintien ● Bit 1 = 0 : Chargement de la configuration via le bornier Bit 1 = 1 : Utilisation d'une configuration Flash EEPROM ● Commande du bus au redémarrage : Bit 2 = 0 : Automatique Bit 2 = 1 : Via l'application ● Commande E/S au redémarrage : Bit 3 = 0 : Automatique Bit 3 = 1 : Via l'application ● Synchronisation de l'échange de données Bit 4 = 0 : Tâche MAST Bit 4 = 1 : Tâche FAST ● Bit 5 réservé ● Chien de garde carte CANopen PCMCIA Bit 6 = 0 : Activé Bit 6 = 1 : Désactivé ● Bits 7 à 15 réservés
%KW0.m.1.2	INT	R	Bits de configuration Taille de la configuration du bus dans la mémoire (en nombre d'octets)

Objet	Type	Accès	Signification
%KW0.m.1.3	INT	R	Bits de configuration Taille de la zone **image d'entrée** dans la mémoire (en nombre de mots)
%KW0.m.1.4	INT	R	Bits de configuration Taille de la zone **image de sortie** dans la mémoire (en nombre de mots)
%KW0.m.1.5	INT	R	Bits de configuration Adresse de début de la zone **image d'entrée** (%MW)
%KW0.m.1.6	INT	R	Bits de configuration Adresse de début de la zone **image de sortie** (%MW)

4.6 Codes d’erreur CANopen

Codes d’erreur CANopen

Présentation

Les tableaux suivants présentent les différents codes d’erreur qui peuvent survenir dans une configuration CANopen.

Les explications données dans chaque tableau vous permettent de programmer l’application pour détecter et corriger plus facilement toute opération future anormale.

Codes d’erreur du module

Le tableau suivant décrit les codes d’erreur localisés dans les mots `%IW0.m.1.1` (dans la colonne Description) et `%IW0.m.1.2` (dans la colonne Détails).

Code	Description	Détails
0	Pas d’erreur, fonctionnement correct	
	Erreurs standard	
100	Adresse incorrecte (pointeur NULL)	-
101	Valeur incorrecte	Valeur
102	Objet ID incorrect	ID
103	Statut pilote incorrect	Code du statut
104	Taille de la zone mémoire de lecture incorrecte	Taille mémoire
105	Taille des données à écrire incorrecte	Taille des données
106	Temporisation	Dépassement du temps pour un transfert SDO ou un statut de chargement lors du chargement d’une configuration
107	Erreur de synchronisation	-
108	Arrêté par l’utilisateur	-
	Erreurs de ressource	
200	Dépassement mémoire	-
201	Ressources mémoire saturées	-
	Erreurs de messagerie	
300	Adresse source incorrecte	Valeur d’adresse
301	Adresse de destination incorrecte	Valeur d’adresse

Code	Description	Détails
302	Service incorrect	Code de service
303	Classe service incorrecte pour une ID segment	Valeur spécifiée
304	Fonction primitive d'un service incorrecte	Code de fonction
305	ID du service appelé incorrecte	Valeur spécifiée
306	Porte de communication incorrecte	Numéro de port
307	ID équipement de bus incorrecte	Valeur de l'ID
308	Index SDO incorrect	Valeur de l'index
309	Sous-index SDO incorrect	Valeur du sous-index
310	Erreur distante lors de l'exécution d'un service	Code d'erreur
311	COB-ID incorrecte	Valeur de la COB-ID
312	Type de transfert sur la couche liaison incorrect	Code du transfert requis : ● 101: envoi ● 102: réception ● 103: envoi et réception
	Erreurs carte PCMCIA	
600	Carte manquante	-
601	Carte différente de la TSX CPP 110 détectée	-
602	La carte n'est pas prête à communiquer	-
603	La carte n'est plus en mode RUN	-
	Erreurs de communication de carte PCMCIA	
700	Erreur lors de l'envoi d'un message à la carte	-
701	Erreur lors de la réception d'un message venant de la carte	-
702	Erreur lors de l'envoi d'un PDO sortie à la carte	-
703	Erreur lors de la réception d'un PDO entrée venant de la carte	-
	Erreurs de configuration	
800	Taille des données de configuration du bus incorrecte	Taille des données de configuration
801	Taille des données de l'image d'entrée incorrecte.	Taille, en nombre de mots, fixée au démarrage de la carte

Code	Description	Détails
802	Taille des données de l'image de sortie incorrecte.	Taille, en nombre de mots, fixée au démarrage de la carte
803	Chevauchement de zones mémoire réservées aux entrées et aux sorties	Type de chevauchement : ● 1: le début de la zone d'entrée couvre la fin de la zone de sortie ● 2: le début de la zone de sortie couvre la fin de la zone d'entrée
804	Zone chargement de données introuvable	Type de zone : ● 1: données globales ● 2: paramètres du bus ● 3: mode synchronisation
805	Somme de contrôle des données de configuration incorrecte (incohérence des données de configuration du bus en mode Sycon)	Somme de contrôle des données de configuration du bus localisée dans la mémoire flash de la carte
806	Rapport négatif sur le chargement de la configuration	Octet de poids fort : code d'erreur carte *(voir page 104)*. Octet de poids faible : ● 16#00: pas de chargement en cours ● 16#01: demande de chargement ● 16#02: chargement en cours ● 16#03: chargement terminé ● 16#11: transfert vers le PC requis ● 16#12: transfert vers le PC en cours

Détails du code 806

Le tableau suivant décrit les codes d'erreur localisés dans l'octet de poids fort du mot `%IW0.m.1.2` et pour la valeur 806 du mot `%IW0.m.1.1`.

Code	Description
48	Temporisation
52	Code de zone inconnu
53	Taille mémoire maximale dépassée
55	Paramètre incorrect.
57	Erreur de séquence lors du chargement
59	Données incomplètes chargées
60	Adresse double
61	La taille du tableau d'adresses PDO est trop grande
62	La taille de la zone paramètre des équipements du bus est trop grande
63	Mode de transmission PDO inconnu
64	La taille des données PDO est trop grande

Code	Description
65	Vitesse de transmission inconnue
66	Synchro COB-ID hors limites
67	Valeur du message Synchro Timer Preset hors limites
68	Taille des données d'entrée + offset plus grands que la taille maximale de la zone d'image d'entrée
69	Taille des données de sortie + offset plus grands que la taille maximale de la zone d'image de sortie
70	Incohérence entre la configuration des PDO et le tableau d'adresse PDO
71	Longueur du tableau d'adresse PDO incorrecte
72	Longueur des données chargées incorrecte
73	Message COB-ID urgent hors limites
74	Message COB-ID de gestion d'un équipement du bus hors limites
75	Indicateur de longueur PDO hors limites
76	La taille des données SDO est trop grande

Codes d'historique

Le tableau suivant décrit les codes d'erreur localisés dans les cinquième et sixième octets des diagnostics du tableau de l'historique.

Code	Description
3	Service rejeté par l'équipement
17	Pas de réponse de l'équipement
51	La longueur de la zone mémoire de réception est trop grande
53	La longueur des données de protocole fragmentées est plus grande que la taille de la mémoire tampon
54	Fonction inconnue requise par le pilote carte
55	Adresse de l'équipement du bus hors normes
57	Erreur de séquence lors d'un transfert fragmenté. L'action est annulée.
200	carte non configurée

Codes de diagnostics esclaves

Le tableau suivant décrit les codes d'erreur localisés dans le septième octet d'un tableau de diagnostics d'un esclave.

Code	Description
30	Erreur lors de la surveillance d'un esclave
31	Changement de statut d'un équipement sur le bus, équipement non disponible
32	Erreur séquence lors d'un protocole de surveillance du bus
33	Pas de réponse pour un PDO configuré
34	Pas de réponse lors de la configuration d'un équipement
35	Profil d'un équipement configuré différent du profil de l'équipement présent sur le bus
36	Type d'équipement configuré différent du type d'équipement présent sur le bus
37	Réponse SDO inconnue
38	Trame reçue plus longue que 8 octets
39	Equipement non scruté ou non arrêté (par exemple en mode Autoclear)

Codes d'erreur du chargement Sycon

Le tableau suivant décrit les codes d'erreur qui peuvent survenir lors du chargement de la configuration ou du microprogramme sur la carte via le pilote X-WAY.

Code	Description
0	Pas d'erreur, fonctionnement correct
Erreurs standard	
8001	Pilote hors service
8002	Code d'événement inconnu du pilote
8003	Code de commande non reconnu par le pilote
8004	Commande refusée
8005	Une autre commande est toujours active
8006	Commande envoyée à un équipement incorrect
Erreurs d'allocation	
8010	Pas d'équipement assigné
8011	Equipement déjà assigné
Erreurs de communication	
8020	Envoi d'une requête service lorsqu'aucun équipement n'est connecté
8021	Initialisation d'une connexion lorsqu'il y a déjà une connexion
8022	Temporisation

Code	Description
8030	Erreur de lecture du statut du pilote
8031	Erreur après l'envoi d'une requête au réseau
8032	Boîte d'envoi toujours occupée
8033	Erreur de réponse réseau
8034	Pas de réponse de la boîte de réception
8035	Erreur de transfert de données E/S
Erreurs d'initialisation du pilote	
8080	Erreurs de paramétrage
8081	Erreurs d'initialisation du pilote général
Erreurs de traitement multitâche	
-1	Tâche mot non créée
-2	Pointeur tâche ou objet synchronisé incorrect
-3	Aucun événement de synchronisation n'a été créé

Exemples d'installation de bus CANopen

5

Objet de ce chapitre

Ce chapitre utilise un exemple pour décrire l'installation d'un bus CANopen avec l'aide de l'outil SyCon (V2.8) et du logiciel Unity Pro.

Contenu de ce chapitre

Ce chapitre contient les sous-chapitres suivants :

Sous-chapitre	Sujet	Page
5.1	Description de l'exemple	110
5.2	Mise en oeuvre matérielle	111
5.3	Mise en oeuvre logicielle	119

5.1 Description de l'exemple

Description de l'exemple

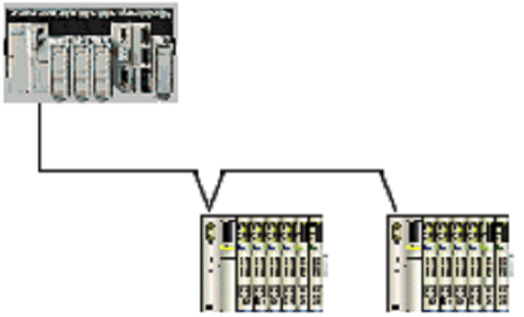
Présentation

Cet exemple est donné à titre didactique. Il vous permettra de suivre les différentes étapes de la configuration d'une architecture CANopen composée :

- d'un module maître **TSX CPP 110** installé dans un automate Premium,
- de deux équipements esclaves **Advantys STB** :
  - Noeud 2 : une interface CANopen STBNCO2212, un module d'alimentation STBPDT3100, un module 4 entrées STBDDI3420 et un module 4 sorties STBDDO3410.
  - Noeud 3 : configuration identique à celle du noeud 2 avec en complément un module deux entrées analogiques STBAVI1270 et un module deux sorties analogiques STBAVO1250.

Illustration

L'illustration ci-dessous schématise l'architecture utilisée dans l'exemple.



Logiciels requis

La mise en oeuvre de cet exemple nécessite les logiciels suivants :

- **Advantys V 1.1**, pour configurer les îlots STB.
- **Sycon V2.8**, pour configurer le bus CANopen.
- **Unity Pro V2.0**, pour configurer l'automate.

5.2 Mise en oeuvre matérielle

Objet de ce sous-chapitre

Ce sous-chapitre présente la mise en oeuvre matérielle de l'exemple CANopen.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Configuration matérielle îlots Advantys	112
Configuration matérielle du maître	115
Configuration matérielle du bus	116

Configuration matérielle îlots Advantys

Présentation

La première phase de mise en oeuvre consiste à configurer matériellement les esclaves CANopen. Après avoir assemblé les différents éléments des STB décrits précédemment puis effectué les raccordements des alimentations, vous devez suivre les étapes suivantes.

Comment fixer la vitesse de transmission

Dans cet exemple nous allons voir comment configurer une vitesse de transmission de 250 KBit/s pour chaque STB.

Etape	Action
1	Mettez le premier STB hors tension.
2	Positionnez le switch du haut sur 4. **Note** : 0 = 10 KBits/s, 1 = 20 KBits/s, 2 = 50 KBits/s, 3 = 125 KBits/s, 4 = 250 KBits/s, 5 = 500 KBits/s, 6 = 800 KBits/s et 7 = 1MBits/s.
3	Positionnez le switch du bas sur "Baud rate" (position supérieure à 9).
4	Mettez le STB sous tension. **Résultat** : le STB est configuré pour travailler à une vitesse de 250 KBit/s.
5	Reprenez les mêmes étapes pour le second STB.

Comment fixer l’adresse CANopen du STB

Dans cet exemple nous allons voir comment configurer l’adresse de chaque STB (2 pour le premier et 3 pour le second).

Etape	Action
1	Mettez le STB hors tension.
2	Positionnez le switch du haut sur 0 (chiffre des dizaines).
3	Positionnez le switch du bas sur 2 (chiffre des unités). **Note** : pour le second STB sélectionnez 3.
4	Mettez le STB sous tension. **Résultat** : le STB est configuré avec l’adresse définie sur les switchs.

Comment charger la configuration du STB

Dans cet exemple nous allons voir comment charger une configuration dans un STB en utilisant la configuration automatique sans carte SIM.

Etape	Action
1	Vérifiez que le STB est sous tension.
2	Enlevez la carte SIM si elle est présente.
3	Appuyez pendant plus de 5 secondes sur le bouton **Reset**. **Résultat** : le STB effectue sa procédure de démarrage et d'initialisation, la configuration matérielle est scrutée (modules du STB) puis stockée dans la mémoire flash. **Note** : si la carte SIM est présente, le STB essaye de charger la configuration qu'elle contient, si la carte SIM n'est pas présente, le STB essaye de charger la configuration contenue dans la mémoire flash. L'appui sur le bouton **Reset** permet de mettre à jour la mémoire flash avec la configuration physique réellement présente. Vous devez toujours effectuer une initialisation lorsque la configuration a changé ou lorsque vous ne connaissez pas le contenu de la mémoire flash.

Comment effectuer une vérification visuelle

Les STB sont prêts à dialoguer avec une carte CANopen TSX CPP 110, une dernière vérification s'impose :

Etape	Action
1	Vérifiez que les voyants **RUN** et **PWR** du module CANopen (NCO) sont allumés.
2	Vérifiez que le voyant **CANRUN** du module CANopen (NCO) clignote.
3	Vérifiez que les voyants **IN** et **OUT** du module d'alimentation (PDT) sont allumés.
4	Vérifiez que le voyant **RDY** des modules d'E/S est allumé pour chacun des modules d'E/S. **Résultat** : le diagnostic visuel à l'aide des voyants permet de penser que les STB sont correctement configurés et prêts à être mis en oeuvre sur un bus CANopen.

Défaut de configuration

Lorsque la configuration présente dans la mémoire flash est différente de la configuration réelle (configuration physique) les voyants :

- du module CAnopen (NCO) **RUN** et **PWR** sont allumés, **CANRUN** clignote en vert, **ERR** et **CANERR** clignotent en rouge.
- du module d'alimentation (PDT) **IN** et **OUT** sont allumés.
- des modules d'E/S **RDY** et **OUT** sont clignotants pour chaque module non présent dans la configuration de la mémoire flash et sont allumés fixe pour les autres.

Défaut module

Certains modules peuvent afficher des situations de défaut (absence d'alimentation des sorties sur un module DDO3230 par exemple). Dans ce cas, **RDY** est allumé et **ERR** clignote sur le module en défaut.

Configuration matérielle du maître

Présentation

Le bus CANopen est géré par un automate Premium dans lequel est installé et configuré une carte TSX CPP 110.

Pour plus de détail sur la carte TSX CPP 110 reportez-vous au chapitre matériel *(voir page 22)* de ce document.

Procédure de montage

Le tableau suivant décrit la marche à suivre pour installer physiquement une carte TSX CPP 110.

Etape	Action
1	Vérifiez que l'alimentation de l'automate est coupée.
2	Montez et fixez les modules du rack de base.
3	Insérez la carte TSX CPP 110 dans son emplacement sur le processeur *(voir page 25)*.
4	Raccordez l'alimentation de l'automate et effectuez la mise sous tension. **Note** : l'automate étant correctement alimenté la configuration logicielle peut maintenant être réalisée.

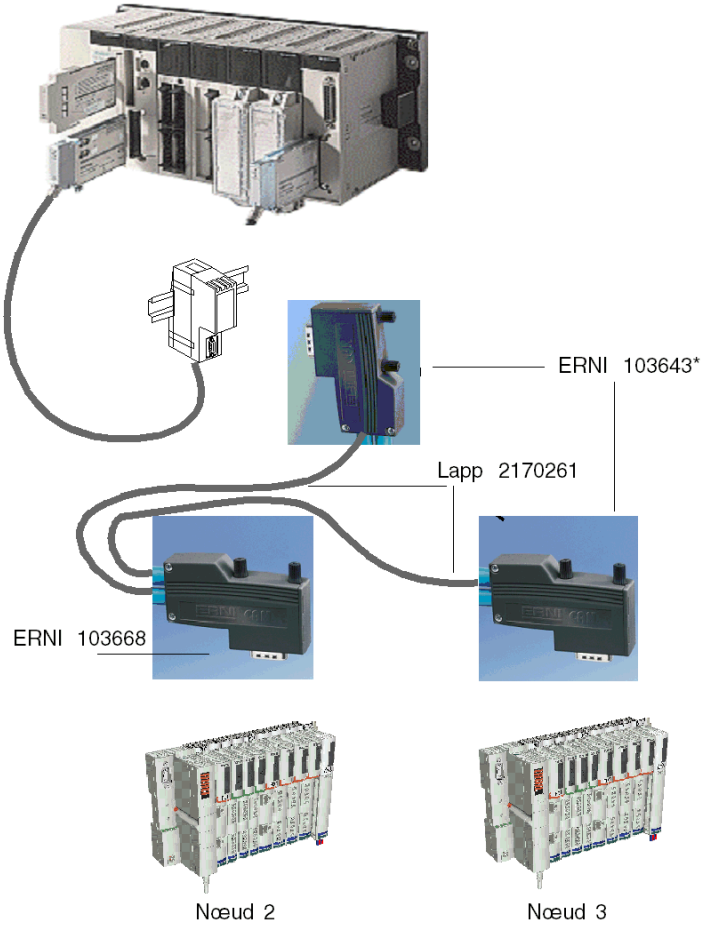
Configuration matérielle du bus

Présentation

Une fois la configuration matérielle pour les esclaves et le maître CANopen terminée, vous devez connecter ces différents équipements.

Exemples de câblage pour le bus

La figure ci-dessous représente une solution de câblage.



* : impédance de ligne de terminaison intégrée (120 Ohms).

Quelques références

Exemple de câble de l'entreprise **Selectron** :

- DCA 701 (code d'élément 44170014)

Pour plus d'informations, voir le site Web : `http://www.selectron.ch`.

Exemple de câble de l'entreprise **Lapp** :

- UNITRONIC BUS CAN 2170261 : Câble double paire torsadée blindée, 120 Ohms.

Pour plus d'informations, voir le site Web :
`http://www.lappcable.com/products`.

Exemple de connecteurs de l'entreprise **ERNI** :

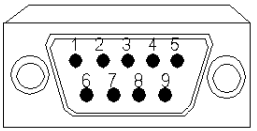
- 1 connecteur de chaînage, référence 103668 (connecté au nœud 2).
- 2 connecteurs d'extrémité de bus, référence 103643 (un d'entre eux est connecté à l'équipement de connexion TSX CPP 110 et l'autre au nœud 3).

Pour plus d'informations, voir le site Web : `http://www.erni.com`.

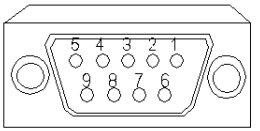
Figure de connexion

Les connecteurs utilisés sont des connecteurs SUB D 9 broches :

Connecteur mâle



Connecteur femelle



Description des broches

Ce tableau montre la fonction de chaque broche.

Elément	Description
1	Réservé
2	CAN_L
3	CAN_GND
4	Réservé
5	NC
6	NC
7	CAN_H
8	Réservé
9	NC

NOTE : Assurez-vous que vous avez connecté correctement le blindage du câble au connecteur.

Pour plus de détails, reportez-vous au document "Manuel d’installation de matériel CANopen" sur telemecanique.com

5.3 Mise en oeuvre logicielle

Objet de ce sous-chapitre

Ce sous-chapitre présente la mise en oeuvre logicielle de l'exemple CANopen.

Les étapes à réaliser sont les suivantes :

- Création de la configuration des STB Advantys et génération des fichiers EDS de chaque noeud.
- Création de la configuration CANopen sous Sycon (utilisation des fichiers EDS précédemment créés pour configurer chaque noeud).
- Création de l'application automate sous Unity Pro et transfert dans le Premium/Atrium.

Contenu de ce sous-chapitre

Ce sous-chapitre contient les sujets suivants :

Sujet	Page
Configuration logicielle Advantys	120
Déclaration du maître CANopen sous Sycon et import EDS	126
Configuration du bus CANopen	129
Déclaration des noeuds 2 et 3	130
Configuration des noeuds 2 et 3	132
Configuration de la carte PCMCIA TSX CPP 110	134
Mise au point	139

Configuration logicielle Advantys

Présentation

La création de la configuration des noeuds Advantys puis la sauvegarde des fichiers EDS s'effectue à l'aide du logiciel Advantys.

Ses principales fonctions sont les suivantes :

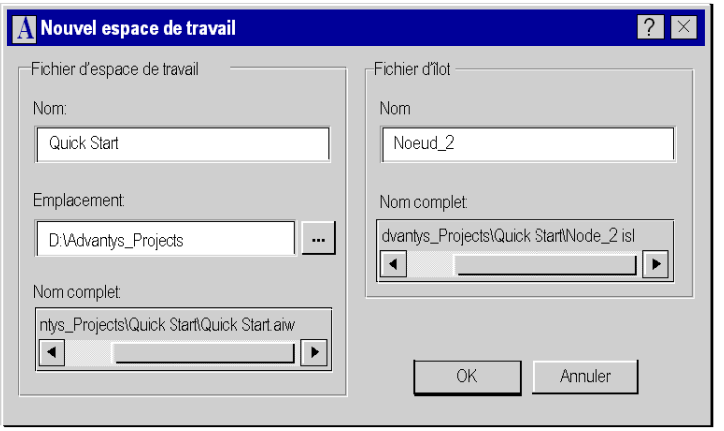
- modification des paramètres par défaut des modules d'E/S (mode de replis des sorties par exemple)
- chargement de la configuration dans la carte SIM (si elle est présente),
- génération des fichiers EDS.

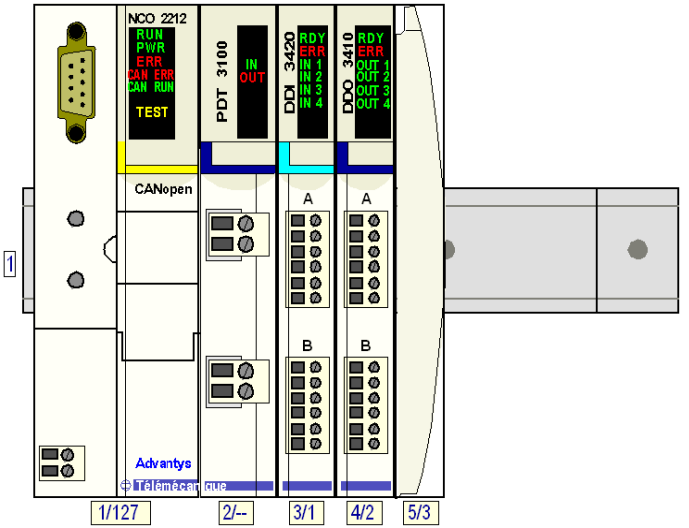
NOTE : dans notre exemple seule la dernière fonction est nécessaire car nous n'utilisons pas de carte SIM et la configuration par défaut nous satisfait pleinement *(voir page 113)*.

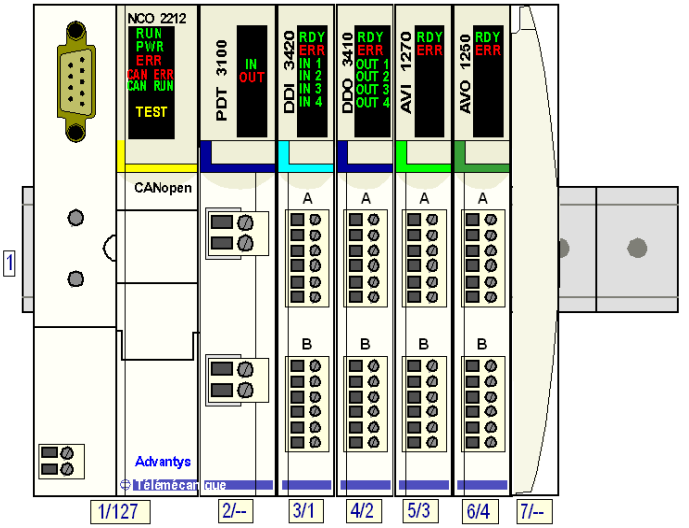
NOTE : le logiciel Advantys n'est pas obligatoire car vous pouvez utiliser les fichiers EDS génériques disponibles sous Sycon. Toutefois ce cas de figure nécessite des connaissances approfondies du logiciel Sycon.

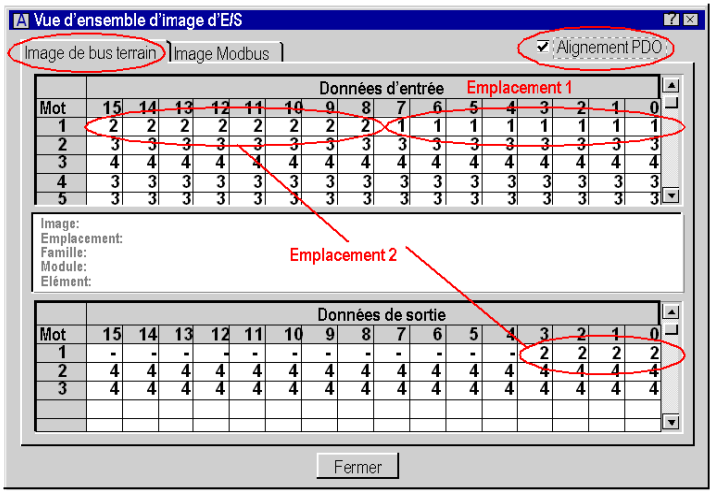
Comment créer les fichiers EDS

Le tableau suivant décrit la marche à suivre pour créer les fichiers EDS utilisables sous Sycon.

Etape	Action
1	Ouvrez le logicel Advantys et créez un nouvel espace de travail (**Fichier** → **Nouvel espace de travail...**) en indiquant : ● le nom du projet (Quick start), ● le chemin de sauvegarde(`D:\Advantys_Projetcs`), ● le nom du noeud (Noeud_2) **Résultat** :  Le nom du fichier projet sera donc **Quick start.aiw**, le nom du fichier de la STB Advantys sera **Noeud_2.isl**. **Note** : tous les noeuds d'un même bus doivent être déclarés dans le même répertoire projet.

Etape	Action
2	Configurez le **Noeud_2** en glissant les modules à partir du catalogue matériel situé à droite de l’écran. **Résultat** :  **Note** : attention à ne pas oublier la terminaison de bus STB XMP 1100.
3	Ajoutez un nouvel ilôt à partir du menu **Fichier** dont le nom sera **Noeud_3**.

Etape	Action
4	Répétez l'étape **2** pour le **Noeud_3**. **Note** : si vous avez oublié l'alimentation ainsi que la terminaison STB XMP 1100 vous ne pouvez pas visualiser la configuration en mode connecté. Vous devez alors la rajouter manuellement. **Résultat** : 
5	Sélectionnez le **Noeud_3** dans le menu **Ilot** →**Aperçu d'image d'E/S** pour visualiser l'image des E/S.

Etape	Action
6	Cliquez sur l'onglet **Image Bus terrain** et cochez la case **Alignement PDO** pour visualiser les éléments du noeud 3. **Résultat** :  **Note** : nous voyons que le noeud 3 possède 3 mots en sorties et 5 mots en entrée. Les entrées de l'emplacement 1 (module DDI) sont situées dans l'octet de poids faible du mot 1. Les E/S du module situé à l'emplacement 2 (module DDO) sont situés dans le mot d'entrée 1 (poids fort) et ses sorties dans le mot de sortie 1, ... Les règles générales de mapping sont expliquées dans la suite de ce document.
7	Répétez l'étape **6** pour le **Noeud_2** afin de visualiser les éléments du noeud 2.
8	Sélectionnez le noeud 2 et créez le fichier EDS en sélectionnant l'option **Fichier →Exporter Noeud_2...** et en l'appelant **Noeud_2**. **Note** : dans notre exemple le fichier sera sauvegardé à l'emplacement suivant : `D:\Advantys_Projects\Quick Start\Noeud_2.eds`.
9	Sélectionnez le noeud 3 et créez le fichier EDS en sélectionnant l'option **Fichier →Exporter Noeud_3...** et en le nomant **Noeud_3**. **Note** : dans notre exemple le fichier sera sauvegardé à l'emplacement suivant : `D:\Advantys_Projects\Quick Start\Noeud_3.eds`.

Règles générales de mapping

Voici les principales règle de positionnement des données dans la mémoire Advantys :

- Le premier bloc est le bloc des entrées Tout ou Rien, ensuite vient le bloc des entrées analogiques.
- Dans chaque bloc, les bits E/S sont classés selon la position physique du module.
- Les bits d'E/S sont triées par leur numéro sur le module. Nous trouverons en premier les valeurs des E/S, ensuite l'echo (pour les sorties seulement) puis les bits d'état.
- Les E/S analogiques sont également triées par leur numéro sur le module. Les valeurs d'E/S analogiques sont situées dans le bloc analogique tandis que les bits de status sont situés dans le bloc des entrées Tout ou Rien.

Exemple des données d'entrée du noeud 3

Données d'entrée				
Mot	bits 15 à 12	bits 11 à 8	bits 7 à 4	bits 3 à 0
1	Bits d'état emplacement 2	Bits d'écho des sorties emplacement 2	Bits d'état emplacement 1	Bits d'entrée emplacement 1
2	Octet d'état emplacement 3		Octet d'état emplacement 3	
3	Octet d'état emplacement 4		Octet d'état emplacement 4	
4	Entrées analogiques emplacement 3			
5	Entrées analogiques emplacement 4			

Exemple des données de sortie du noeud 3

Données de sortie				
Mot	bits 15 à 12	bits 11 à 8	bits 7 à 4	bits 3 à 0
1	-	-	-	Bits de sortie emplacement 2
2	Sorties analogiques emplacement 4			
3	Sorties analogiques emplacement 4			

NOTE : l'emplacement est le chiffre de droite sur l'étiquette qui apparaît en bas de chaque module.

Déclaration du maître CANopen sous Sycon et import EDS

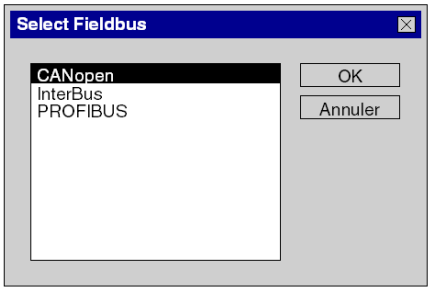
Présentation

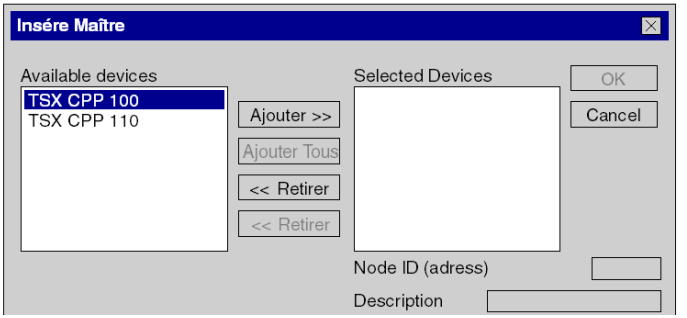
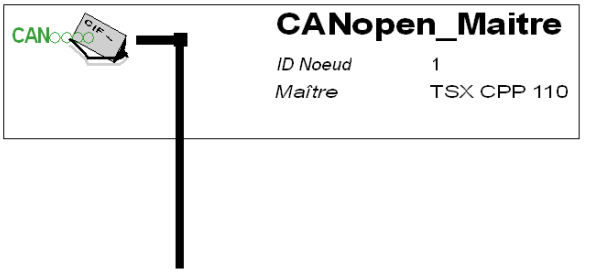
Nous allons maintenant utiliser le logiciel Sycon pour créer le bus CANopen et générer la description du bus qui sera chargée dans l'automate Premium à l'aide du logiciel Unity Pro.

Dans un premier temps nous allons déclarer le maître du bus puis importer les fichiers EDS des noeuds 2 et 3 afin qu'ils soient reconnus par le logiciel Sycon.

Comment déclarer le maître du bus CANopen

Le tableau ci-dessous présente les différentes étapes pour déclarer le maître CANopen.

Etape	Action
1	Lancez l'outil Sycon à partir du menu **Démarrer** de Windows ou bien, depuis l'écran de configuration de la carte TSX CPP 110 (dans le logiciel Unity Pro), cliquez sur l'icône **Résultat :** L'outil SyCon apparaît à l'écran.
2	Sélectionnez la commande **Fichier** →**Nouveau** **Résultat :** L'écran suivant apparaît : 
3	Sélectionnez **CANopen** puis validez par **Ok**. **Résultat :** Une architecture vide apparaît à l'écran.

Etape	Action
4	Sélectionnez la commande **Insérer →Maître**. **Résultat :** L'écran suivant apparaît : 
5	• Sélectionnez **TSX CPP 110**, • cliquez sur **Ajouter**, • entrez un nom représentant l'équipement maître dans le champ **Description** , **Note :** Le nom ne doit pas contenir d'espace ni de caractère accentué et il est limité à 32 caractères maximum. • validez par **Ok**. **Résultat :** L'architecture suivante apparaît : 
6	Enregistrez le projet CANopen sous le nom Demo_cfg.co. **Note** : notez bien l'endroit où est stocké le fichier .CO car vous devrez le réutiliser pour le charger dans la configuration de la carte sous Unity Pro.

Comment importer les fichiers EDS

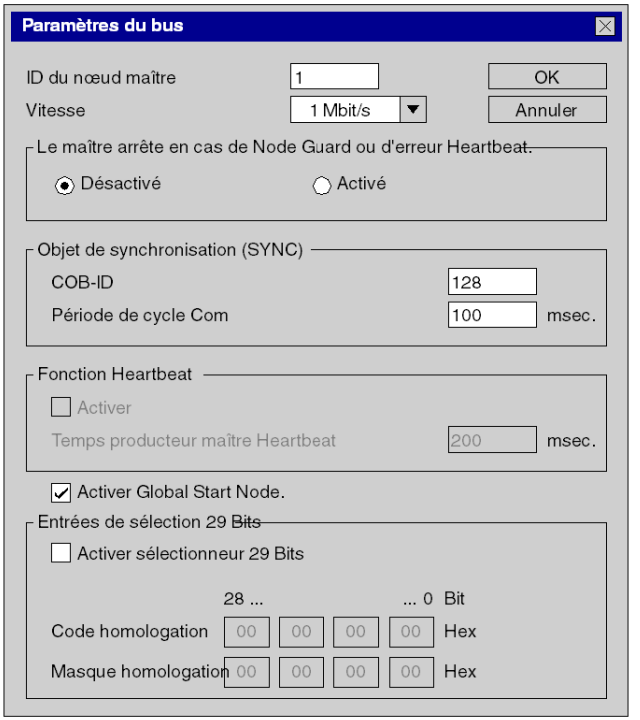
Le tableau ci-dessous présente les différentes étapes pour importer les fichiers EDS sous Sycon.

Etape	Action
1	Lancez le logiciel Sycon si cela n'a pas été déjà fait pour déclarer le maître du bus *(voir page 126)*. **Résultat :** L'outil SyCon apparaît à l'écran.
2	Sélectionnez l'option **Fichier** →**Copier EDS** pour importer les fichiers EDS sous Sycon.
3	Sélectionnez le fichier **Noeud_2.eds** du répertoire `D:\Advantys_Projects\Quick Start` pour importer le fichier EDS du noeud 2. **Note** : refusez l'import du bitmap car ce fichier n'a pas été créé.
4	répétez les étapes **2** et **3** pour le noeud 3 avec le fichier **Noeud_3.eds** du répertoire `D:\Advantys_Projects\Quick Start` pour importer le fichier EDS du noeud 3. **Note** : refusez l'import du bitmap car ce fichier n'a pas été créé. **Résultat** : les deux fichiers EDS sont maintenant disponibles pour l'insertion des deux esclaves CANopen *(voir page 130)*.

Configuration du bus CANopen

Procédure

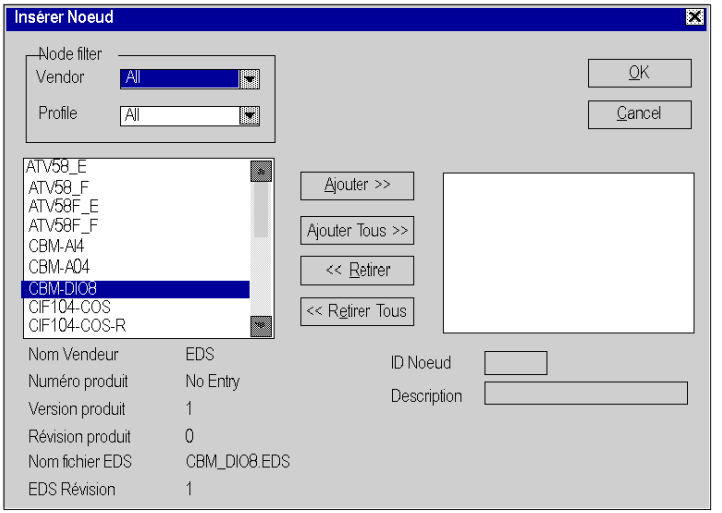
Le tableau ci-dessous répertorie les différentes étapes de déclaration du bus CANopen.

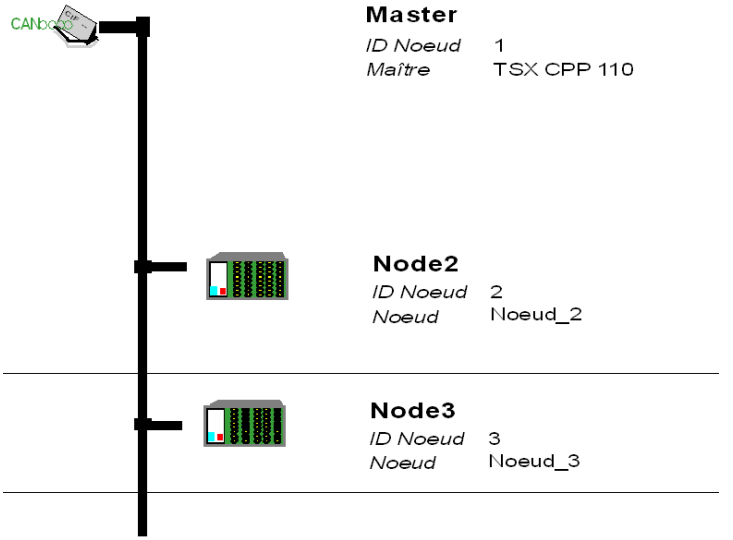
Etape	Action
1	Sélectionnez la commande **Paramètres →Paramètres du Bus**. **Résultat :** l'écran suivant apparaît : 
2	Configuration : • Vitesse à 250 Kbit/s, • La valeur de **SYNC COB-ID** à 128 (valeur par défaut), • Période de cycle Com à 100 ms.
3	Sélectionnez **Désactiver** pour **arrêter le maître en cas de Node Guard ou d'erreur Heartbeat**.
4	Sélectionnez **Activer Global Start Node**.
5	Validez par **Ok**.

Déclaration des noeuds 2 et 3

Marche à suivre

Le tableau ci-dessous présente les différentes étapes pour déclarer les noeuds 2 et 3.

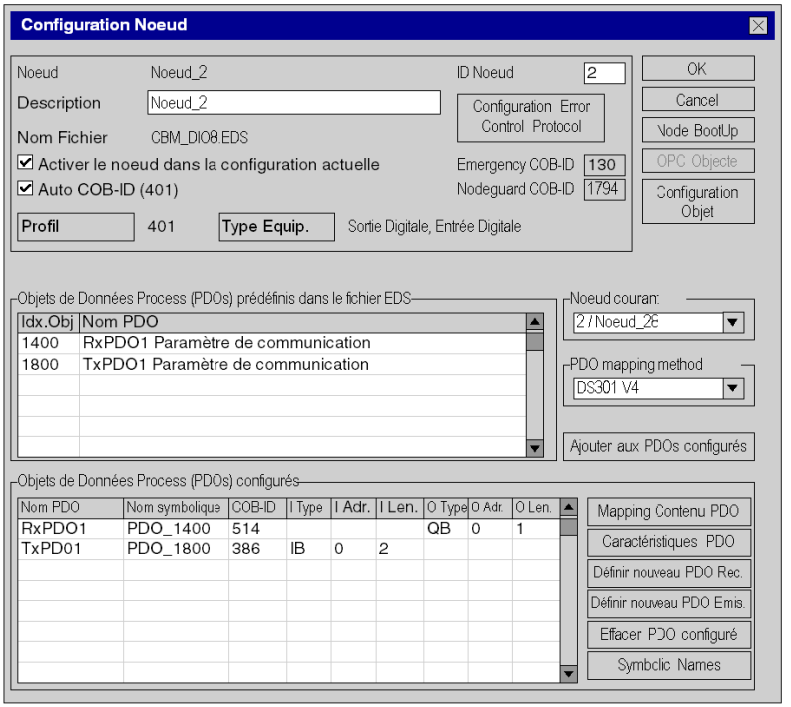
Etape	Action
1	Sélectionnez la commande **Insérer →Noeud**. **Résultat :** Un curseur apparaît :
2	Placer le curseur sur le bus en dehors du cadre délimitant le maître et effetuez un simple clic. **Résultat :** L'écran suivant apparaît : 
3	Sélectionnez **Noeud_2** dans la liste des équipements disponibles.
4	Cliquez sur le bouton **Ajouter** pour insérer le **Noeud_2** dans la liste des équipements sélectionnés.
5	Sélectionnez **Noeud_3** dans la liste des équipements disponibles.

Etape	Action
6	Cliquez sur le bouton **Ajouter** pour insérer le **Noeud_3** dans la liste des équipements sélectionnés et validez par **OK**. **Résultat :** L’architecture suivante apparaît : 

Configuration des noeuds 2 et 3

Comment configurer le noeud 2

Le tableau ci-dessous présente les différentes étapes pour configurer le noeud 2.

Etape	Action
1	Effectuez un double clic sur le noeud 2. **Résultat :** L'écran de configuration apparaît, il affiche la grille des deux PDO (Predefined Process Objects) du noeud. Un PDO de réception (RxPDO) qui permet de configurer les sorties et un PDO d'émission (TxPDO) qui permet de configurer les entrées.
2	Effectuez un double clic sur la première ligne (PDO RxPDO1) pour afficher la fenêtre de caractéristiques du PDO.
3	Validez par **OK** car nous utiliserons la configuration par défaut. Résultat : le PDO apparaît maintenant dans la zone **Objets de données Process (PDOs) configurés**.
4	Répétez les étapes **2** et **3** pour le second PDO. **Résultat :** vous devez obtenir l'écran configuré comme suit : 
5	Cliquez sur OK pour valider la configuration du noeud 2. **Note** : les PDO étant configurés, le logiciel Sycon possède maintenant toutes les informations de configuration du noeud 2 afin d'utiliser le fichier EDS correspondant.

Comment configurer le noeud 3

Le tableau ci-dessous présente les différentes étapes pour configurer le noeud 3.

Etape	Action
1	Effectuez un double clic sur le noeud 3. **Résultat :** L'écran de configuration apparaît, il affiche la grille des quatre PDO (Predefined Process Objects) du noeud. Deux PDO de réception qui permettent de configurer les sorties et deux PDO d'émission qui permettent de configurer les entrées.
2	Effectuez un double clic sur la première ligne pour afficher la fenêtre de configuration du premier PDO.
3	Validez par **OK** car nous utiliserons la configuration par défaut.
4	Répétez les étapes **2** et **3** pour les 3 autres PDO.
5	Cliquez sur le bouton **Configuration objet** pour activer la transmission des entrées analogiques. **Note** : par défaut la transmission des entrées analogiques est désactivée il est donc nécessaire d'activer cette fonctionnalité.
6	Faites défiler jusqu'à la fin l'ascenseur de la zone **Objets prédéfinis spécifiés dans le fichier EDS** et double cliquez sur la ligne **6423 0 Analog Input Global Enable** dans la liste des objets prédéfinis supportés.
7	Dans la colonne **Valeur choisie** saisissez la valeur **1**.
8	Cliquez sur **OK** pour valider la configuration des objets.
9	Cliquez sur **OK** pour valider la configuration du noeud 3. **Résultat** : le noeud 3 est maintenant configuré. Si vous avez effectué également la configuration du noeud 2 vous pouvez sauvegarder le projet Sycon qui est maintenant utilisable dans l'automate sous le nom de `Demo_cfg.co`.

Configuration de la carte PCMCIA TSX CPP 110

Vue d'ensemble

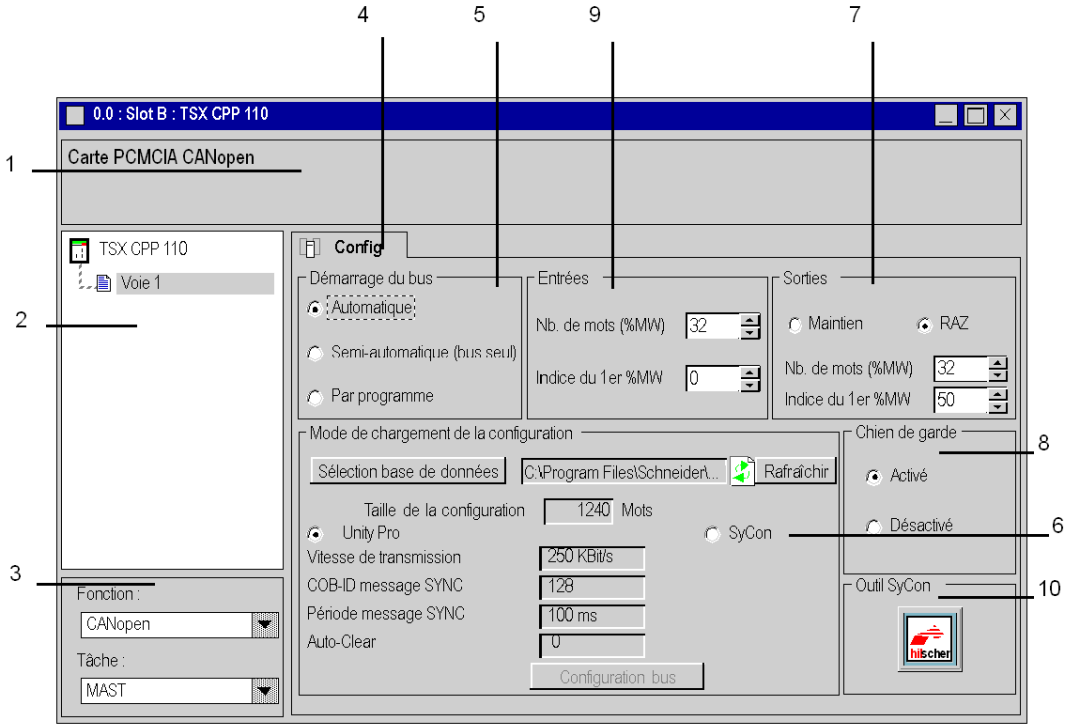
Après avoir créé les fichiers EDS et le bus CANopen à l'aide du logiciel Sycon, vous devez déclarer puis configurer la carte TSX CPP 110 dans l'automate.

Pour déclarer la carte vous pouvez vous reporter au paragraphe correspondant *(voir page 37)* dans cette documentation.

La configuration de la carte dans cet exemple est illustrée sur la figure ci-dessous. Pour plus d'informations sur les configurations possibles, veuillez vous reporter au paragraphe correspondant. *(voir page 134)*.

Illustration

L'écran suivant indique les paramètres de configuration nécessaires à la mise en œuvre de notre exemple :



Eléments et fonctions

Le tableau suivant décrit les différentes zones de l'écran de configuration :

Zone	Numéro	Fonction
Module	1	Cette zone est constituée de l'intitulé abrégé de la carte PCMCIA que vous avez déclarée, il doit s'agir d'une carte CANopen.
Voie	2	Cette zone permet de sélectionner la voie de communication à configurer. Pour obtenir l'onglet de configuration vous devez cliquer sur la voie.
Paramètres généraux	3	Dans cette zone, sélectionnez la tâche associée aux E/S situées sur le bus CANopen. La tâche sélectionnée déterminera le taux d'acquisition des entrées et actualisera les sorties des esclaves de bus CANopen. Dans notre cas nous sélectionnerons la tâche maître (MAST).
Onglet	4	L'onglet au premier plan indique le type d'écran affiché. Dans notre cas il s'agit de l'écran de configuration. Si vous êtes en mode connecté, vous aurez accès à d'autres onglets. Dans ce cas, vérifiez que vous avez cliqué sur l'onglet **Config.** pour obtenir l'écran présenté dans l'illustration.

Zone	Numéro	Fonction
Config	5	Cette zone permet de sélectionner le comportement du bus au démarrage. Sélectionnez **Automatique**.
	6	Cette zone permet de configurer le bus. Cliquez sur le bouton Unity Pro pour que la configuration soit accessible sous Unity Pro. Sélectionnez le fichier `Demo_cfg.co` que vous avez créé avec le logiciel Sycon. **Résultat** : les paramètres du bus sont affichés lorsqu'un fichier .CO est sélectionné. **Remarque** : si vous modifiez le fichier .CO sous Sycon, cliquez sur le bouton **Rafraîchir** afin de le recharger dans la configuration. **Remarque** : si vous désirez visualiser la liste des esclaves du bus cliquez sur le bouton **Configuration bus**.
	7	Cette zone permet de configurer le mode de repli des sorties des équipements du bus ainsi que l'adresse (mémoire interne de l'automate) ou seront lues périodiquement les sorties des équipements CANopen. Dans notre exemple nous sélectionnons **RAZ** et une quantité de **32 mots** débutant à l'adresse **50** (%MW50 à %MW81. Nous verrons un peu plus tard que ces adresses correspondent à des objets topologiques (qui dépendent de l'esclave dans lequel elles sont situées).
	8	Cette zone permet d'activer ou de désactiver le chien de garde du bus CANopen. Le chien de garde est activé par défaut. Il est déclenché lorsque la carte PCMCIA ne peut plus gérer correctement le bus. Lorsqu'il est déclenché, il fait passer les sorties des esclaves à zéro. Dans notre exemple nous allons le laisser activé.
	9	Cette zone permet de configurer l'adresse (mémoire interne de l'automate) ou seront copiées périodiquement les entrées des équipement CANopen. Dans notre exemple nous allons garder les valeurs par défaut 32 et 0 pour indiquer que les entrées seront stockées dans les mots %MW0 à %MW31. Un adressage topologique est également disponible comme pour les sorties.
	10	Ce bouton permet de lancer le logiciel **Sycon** s'il est installé sur le PC.

⚠ AVERTISSEMENT

COMPORTEMENT INATTENDU DE L'APPLICATION

Avant de désactiver le chien de garde, assurez-vous que la carte PCMCIA ne gère pas le bus CANOpen, et que le comportement des équipements reste acceptable.

Le non-respect de ces instructions peut provoquer la mort, des blessures graves ou des dommages matériels.

Validation et génération

Après avoir renseigné les différents champs de l'écran de configuration vous devez :

- valider la configuration,
- générer le projet.

Lorsque ces deux opérations sont effectuées vous pouvez enregistrer votre fichier .STU et transférer votre projet dans l'automate.

Répartition des E/S

Les entrées et les sorties configurées dans le fichier .CO sont réparties de la manière suivante :

- %MW0 contient les entrées du nœud 2.
- %MW50 contient les sorties du nœud 2.
- %MW1 à %MW5 contiennent les entrées du nœud 3.
- %MW51 à %MW53 contiennent les sorties du nœud 3.

Répartition topologique des E/S

Si vous cliquez sur le bouton **Configuration bus** de l'écran de configuration vous pouvez visualiser les deux nœuds du bus CANopen et ainsi obtenir l'adressage topologique des E/S.

Pour le nœud 2 nous avons les entrées suivantes :

- %IW\3.2\0.0.0.0 qui correspond à %MW0.

et les sorties suivantes :

- %QW\3.2\0.0.0.0 qui correspond à %MW50.

Pour le nœud 3 nous avons les entrées suivantes :

- %IW\3.3\0.0.0.0 qui correspond à %MW1,
- %IW\3.3\0.0.0.1 qui correspond à %MW2,
- %IW\3.3\0.0.0.2 qui correspond à %MW3,
- %IW\3.3\0.0.0.3 qui correspond à %MW4,
- %IW\3.3\0.0.0.4 qui correspond à %MW5.

et les sorties suivantes :

- %QW\3.3\0.0.0.0 qui correspond à %MW51,
- %QW\3.3\0.0.0.1 qui correspond à %MW52,
- %QW\3.3\0.0.0.2 qui correspond à %MW53.

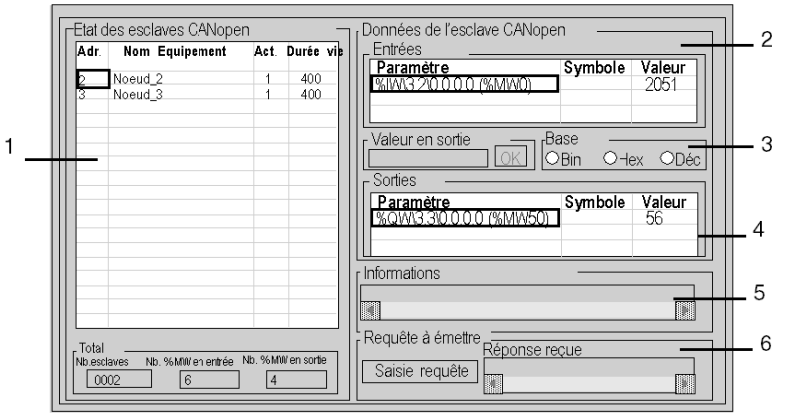
Mise au point

Présentation

L'écran de mise au point est accessible en mode connecté uniquement. Il permet de visualiser le fonctionnement du bus, des esclaves et du maître CANopen.

Illustration

La figure ci-dessous est extrait de l'écran de mise au point de notre exemple.



Description

Le tableau ci-dessous présente les différentes zones de l'écran de mise au point :

Repère	Element	Fonction
1	Etat des esclaves CANopen	Cette zone affiche tous les esclaves (appelés aussi noeuds) du bus CANopen. Un esclave en défaut est affiché en rouge, lorsque le défaut disparaît, il est affiché en bleu, sinon il est affiché en noir. La sélection d'un esclave met à jour les zones 2, 4, 5 et 6. **Act.** : indique si l'esclave a été activé dans la configuration Sycon (1 activé, 0 désactivé) **Durée de vie (Life T.)** : Life Time. Dans notre exemple nous visualisons donc deux esclaves (ou deux noeuds), ce sont les deux noeuds de notre configuration Sycon. **Note** : lorsqu'un noeud est visualisé en rouge vous pouvez obtenir les causes d'erreur en cliquant dessus. La ligne information est automatiquement mise à jour avec les informations de diagnostic.

Repère	Element	Fonction
2	Entrées	Lorsqu'un esclave est sélectionné, cette zone contient la liste des mots qui lui sont associés en entrée. Pour plus de clarté, l'affichage de l'adressage topologique CANopen *(voir page 35)* (%IW\3.2\0.0.0.0) et de la zone mémoire réservée dans l'automate (%MW0) est proposé. Si le noeud 2 est sélectionné nous visualisons l'illustration précédente, si le noeud 3 est sélectionné, l'écran change et visualise les %IW du noeud 3.
3	Valeur en sortie	Lorsqu'un mot de sortie est sélectionné dans la zone 4, il est possible de modifier sa valeur en saisissant une nouvelle valeur puis en cliquant sur le bouton **OK**.
4	Sorties	Lorsqu'un esclave est sélectionné, cette zone contient la liste des mots qui lui sont associés en sortie. Pour plus de clarté, l'affichage de l'adressage topologique CANopen *(voir page 35)* (%QW\3.2\0.0.0.0) et de la zone mémoire réservée dans l'automate (%MW50) est proposé. Si le noeud 2 est sélectionné nous visualisons l'illustration précédente, si le noeud 3 est sélectionné, l'écran change et visualise les %QW du noeud 3.
5	Informations sur ...	Lorsqu'un esclave est sélectionné (cliquez dans la zone 1), cette zone contient son dernier message de diagnostic et pour obtenir des informations sur la carte TSX CPP 110 il suffit de cliquer sur l'entête du tableau.
6	Requête à émettre	Cette zone permet d'émettre une requête SDO. La syntaxe des paramètres est identique à celle utilisée pour effectuer des transferts SDO par les requêtes READ_VAR et WRITE_VAR *(voir page 57)*. L'appui sur le bouton **Saisie requête** fait apparaître les zones de saisie le la requête.

Annexes

Exemple de configuration pour équipements sur le bus CANopen

A

Objet de ce chapitre

Ce chapitre présente trois exemples de configuration pour les équipements sur le bus CANopen.

- un variateur Altivar ATV 58,
- un lecteur Lexium,
- une situation qui requiert plus de 4 PDO par équipement.

Contenu de ce chapitre

Ce chapitre contient les sujets suivants :

Sujet	Page
Configuration d'un variateur de vitesse Altivar	144
Configuration d'un variateur Lexium	147
Configuration de plus de 4 PDO par nœud	150

Configuration d'un variateur de vitesse Altivar

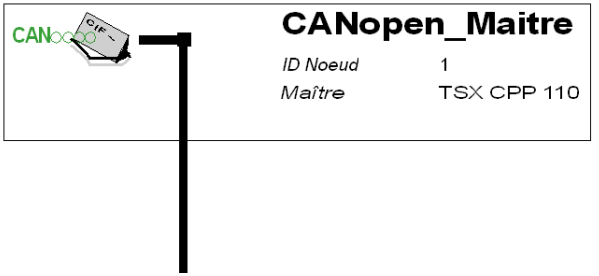
Présentation

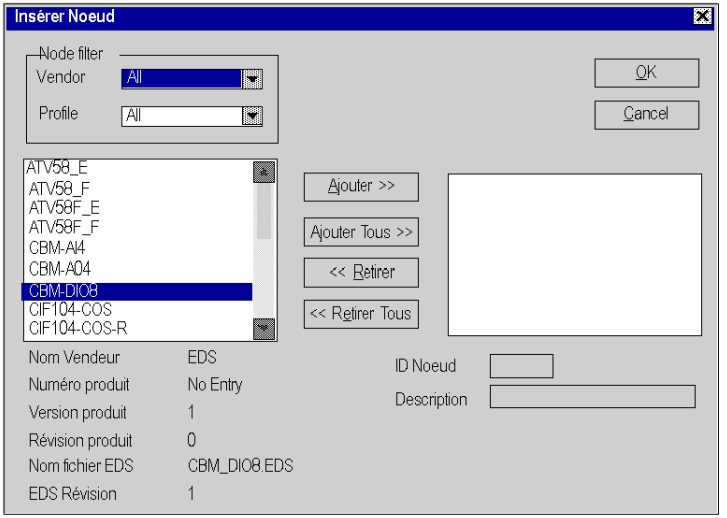
Vous pouvez très facilement configurer un Altivar sur le bus CANopen. Nous allons voir comment réaliser cette configuration.

Avant de lire les paragraphes qui suivent nous vous conseillons d'avoir réalisé l'exemple de mise en oeuvre des STB Advantys sur un bus CANopen *(voir page 109)* décrit dans ce manuel ou bien d'avoir lu ces pages de mise en oeuvre car nous nous y référerons plusieurs fois.

Comment configurer un Altivar

Le tableau ci-dessous présente les différentes étapes pour configurer un Altivar.

Etape	Action
1	Lancez le logiciel Sycon et créez un nouveau projet CANopen *(voir page 126)*. **Résultat :** vous devez obtenir une architecture de bus de ce type : 
2	Sélectionnez la commande **Insérer →Noeud**. **Résultat :** Un curseur apparaît :

Etape	Action
3	Placer le curseur sur le bus en dehors du cadre délimitant le maître et effetuez un simple clic. **Résultat :** L'écran suivant apparaît : 
4	Sélectionnez le fichier EDS de l'Altivar que vous désirez incorporer au bus et cliquez sur **Ajouter**. **Note :** Quatre Altivar sont disponibles : • ATV58_E : fichiers EDS en anglais, • ATV58_F : fichiers EDS en français, • ATV58F_E : fichiers EDS en anglais, • ATV58F_F : fichiers EDS en français,
5	Validez votre choix par **OK**. **Résultat** : l'architecture de bus fait apparaître un nouveau noeud composé d'un variateur. Vous devez maintenant configurer les PDO de ce variateur.
6	Double cliquez sur le noeud du variateur pour faire apparaître la fenêtre des PDO. **Résultat** : un écran similaire à celui des STD Advantys *(voir page 144)* apparaît vous permettant de configurer les deux PDO du variateur : • PDO en émission : 1 mot d'état et 1 mot pour la vitesse courante, • PDO en réception : 1 mot du registre de commande et 1 mot pour la vitesse commandée.

Etape	Action
7	Double cliquez sur chaque PDO et validez les paramètres de transmission par défaut.
8	Vérifiez que les vitesses du noeud et son adresse correspondent à ceux que vous avez fixé dans l'Altivar. **Note** : si l'Altivar est le dernier élément du bus n'oubliez pas d'activer la terminaison de fin bus en positionnant le switch situé sur la carte de communication CANopen de l'Altivar.
9	Cliquez sur OK pour valider la configuration du noeud.
10	Sauvegardez le fichier .CO.
11	Ouvrez votre projet Unity Pro qui contiendra le maître du bus CANopen (carte TSX CPP 110), déclarez la carte, importez le ficher .CO sauvegardé, générez le projet et transférez le dans l'automate comme pour l'exemple STD Advantys *(voir page 134)*. **Résultat** : le bus CANopen et votre Altivar sont opérationnels.

Tests

Vous pouvez effectuer quelques tests afin de contrôler que l'Altivar et le bus fonctionnent correctement en ouvrant l'écran de mise au point de la carte TSX CPP 110 et en effectuant les actions suivantes :

- Vous pouvez démarrer le moteur en saisissant les valeurs 6, 7, 15 dans le registre de commande (premier mot de sortie).
- Vous pouvez fixer la vitesse de rotation du moteur en modifiant la valeur du second mot de sortie.
- Si l'Altivar passe en défaut, vous devez effectuer une réinitialisation en saisissant la valeur 128 dans le registre de commande (premier mot de sortie) puis reprendre les séquences 6, 7 et 15.

NOTE : après avoir vérifié que votre Altivar était bien commandé par la modification de mots de l'automate qui lui sont transmis automatiquement sur le bus CANopen vous pouvez mettre en oeuvre votre projet d'automatisme.

ATV 31

Le variateur de vitesse ATV 31 est également compatible CANopen, pour le détail de sa configuration vous pouvez vous référer au manuel ATV 31, Protocole de communication CANopen.

Configuration d'un variateur Lexium

Présentation

Vous pouvez très facilement configurer un variateur Lexium sur le bus CANopen. Nous allons voir comment réaliser cette configuration.

Avant de lire les paragraphes qui suivent nous vous conseillons d'avoir réalisé l'exemple de mise en oeuvre des STB Advantys sur un bus CANopen *(voir page 109)* décrit dans ce manuel ou bien d'avoir lu ces pages de mise en oeuvre car nous nous y référerons plusieurs fois.

Raccordement du Lexium

Le raccordement du Lexium u bus CANopen s'effectue par l'intermédiaire de l'adaptateur AMO 2CA 001 V000 :

- vissé sur le Lexium puis
- connecté aux câbles du bus CANopen.

NOTE : pour la mise en eouvre de toute autre connectique, reportez-vous au manuel **Lexium CANopen** que vous pouvez trouver sur le **CD ROM Motion Tools 3.0**.

Description des PDO du Lexium

Comme pour l'Altivar, les PDO permettent une gestion implicite du Lexium par l'intermédiaire de mots de l'automate (%MW ou adressage topologique %IW et %QW *(voir page 35)*).

Le Lexium utilise deux types de PDO :

- Les PDO prédéfinis :
  - Ces PDO sont associés à des commandes spécifiques du lexium. Exemple : le PDO 22 de cette annexe est associé à la requête 16#6040 pour le mot de commande et à la requête 16#2060 pour la **Feed rate or current**.
- Les PDO libres :
  - Ces PDO ne sont pas associés à des commandes spécifiques. Leur configuration s'effectue par l'application. Nous ne décrirons pas ces PDO dans cet exemple, vous pouvez vous reporter au manuel Lexium CANopen pour plus de détails.

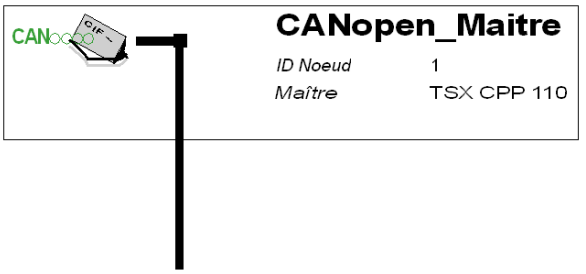
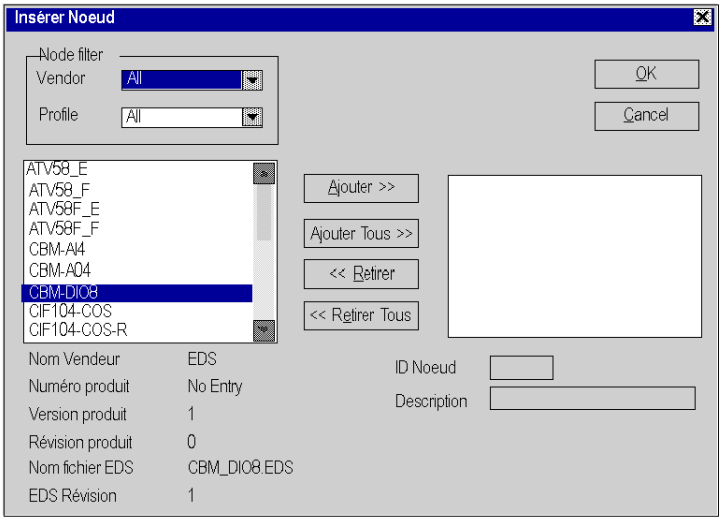
Les PDO sont échangés par deux voies prédéfinies :

- 3 voies de réception (16#2600, 16#2601, 16#2602)
- 3 voies d'émission (16#2A00, 16#2A01, 16#2A02).

NOTE : la taille maximum des données est de 8 ctets par voie et le Lexium doit être de version supérieure ou égale à 5.51.

Comment configurer un PDO du Lexium

Le tableau ci-dessous présente les différentes étapes pour configurer un exemple de PDO en réception, le PDO 22 du Lexium, sur la voie 1. .

Etape	Action
1	Lancez le logiciel Sycon et créez un nouveau projet CANopen *(voir page 126)*. **Résultat :** vous devez obtenir une architecture de bus de ce type : 
2	Sélectionnez la commande **Insérer →Noeud**. **Résultat :** Un curseur apparaît :
3	Placer le curseur sur le bus en dehors du cadre délimitant le maître et effetuez un simple clic. **Résultat :** L'écran suivant apparaît : 
4	Sélectionnez le fichier EDS LEXIUM17D et cliquez sur **Ajouter**.
5	Validez votre choix par **OK**. **Résultat** : l'architecture de bus fait apparaître un nouveau noeud composé d'un Lexium 17D. Vous devez maintenant configurer le PDO 22 de ce variateur.

Etape	Action
6	Double cliquez sur le noeud du variateur pour faire apparaître la fenêtre des PDO. **Résultat** : un écran similaire à celui des STB Advantys *(voir page 148)* apparaît vous permettant de configurer les PDO du variateur :
7	Sélectionnez sur le premier PDO en réceptiont.
8	Cliquez sur le bouton **Ajouter aux PDOs configurés** et validez les paramètres de transmission par défaut en cliquant sur **OK**. **Résultat** : une nouvelle ligne apparaît dans la liste des PDO configurés.
9	Cliquez sur cette ligne puis sur le bouton **Mapping contenu PDO**. **Résultat** : une nouvelle feêtre apparaît.
10	Sélectionnez l'Idx Obj. 2060 (Feed rate or current).
11	Cliquez sur le bouton **Ajouter Objet** puis sur **OK** pour valider. **Résultat** : les requêtes 16#6040 et 16#2060 sont associées au PDO. Vous devez maintenant configurer le PDO.
12	Sélectionnez à nouveau le premier PDO en réception puis cliquez sur le bouton **Configurer Objet**. **Résultat** : La fenêtre de configuration du PDO apparaît.
13	Faites défiler la liste des **Objets prédéfinis spécifiés dans le fichier EDS** et sélectionnez la ligne **1st receive PDO select** de l'objet prédéfini 2600.
14	Cliquez sur le bouton **Ajoute aux obj configurés**. **Résultat** : l'objet apparaît dans la fenêtre des **Objets configurés automatiquement au démarrage du noeud**.
15	Saisissez 22 dans la case correspondante de la colonne **Valeur choisie**.
16	Sauvegardez le ficchier .CO afin de pouvoir l'importer dans la configuration de la carte TSX CPP 110 sous Unity Pro.
17	Ouvrez votre projet Unity Pro qui contiendra le maître du bus CANopen (carte TSX CPP 110), déclarez la carte, importez le ficher .CO sauvegardé, configurez la carte avec les valeurs par défaut, générez le projet et transférez le dans l'automate comme pour l'exemple STB Advantys *(voir page 134)*. **Résultat** : le bus CANopen et votre Lexium sont opérationnels.

Tests

Vous pouvez effectuer quelques tests afin de contrôler que le Lexium et le bus fonctionnent correctement en ouvrant l'écran de mise au point de la carte TSX CPP 110 et en effectuant les actions suivantes :

- Si vous avez utilisé la configuration par défaut de la carte TSX CPP 110, le mot %MW32 (premier mot de sortie) correspond au mot de contrôle du Lexium, vous pouvez le modifier et contrôler la conséquence sur le Lexium.
- De même, le double mot %MD32 vous permet d'accéder à la vitesse.

NOTE : après avoir vérifié que votre Lexium était bien commandé par la modification de mots de l'automate qui lui sont transmis automatiquement sur le bus CANopen vous pouvez mettre en oeuvre votre projet d'automatisme.

Configuration de plus de 4 PDO par nœud

Présentation

Pour chaque nœud, 4 POD de transmission et 4 PDO de réception peuvent être configurés. Cela vient du fait que le logiciel Sycon génère automatiquement les COB-ID pour garantir leur unicité et utilise un algorithme respectant le profil 301. Le paragraphe suivant explique cette opération.

Pour pouvoir utiliser plus de 4 PDO dans chaque direction, vous devez utiliser une procédure manuelle telle que celle décrite pour créer un 5 ^{e} PDO.

Affectation des COB-ID

Les COB-ID possibles se situent entre les valeurs 385 et 1407 (16#180 et 16#57F).

Vous pouvez utiliser le COB-ID de votre choix pour chaque PDO ; toutefois, les COB-ID doivent être uniques dans cet intervalle de valeurs. Les outils de configuration CANopen tel que Sycon sont configurés par défaut pour donner un COB-ID qui respecte ces contraintes.

Le tableau suivant décrit l'algorithme utilisé par le logiciel Sycon pour réaliser ces affectations COB-ID.

PDO	ID	Nœud 1 (décimal)	Nœud 2 (décimal)	...	Nœud 127 (décimal)
1.TXPDO	16#180+ID-Nœud	385	386	...	511
1.RXPDO	16#200+ID-Nœud	513	514	...	639
2.TXPDO	16#280+ID-Nœud	641	642	...	767
2.RXPDO	16#300+ID-Nœud	769	770	...	895
3.TXPDO	16#380+ID-Nœud	897	898	...	1023
3.RXPDO	16#400+ID-Nœud	1025	1026	...	1151
4.TXPDO	16#480+ID-Nœud	1153	1154	...	1279
4.RXPDO	16#500+ID-Nœud	1281	1282	...	1407

Lorsque le COB-ID détermine la priorité du message (plus petite est la valeur, plus grand est sa priorité), les conséquences sont les suivantes :

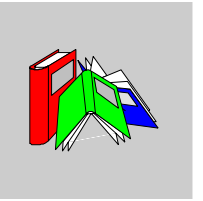
- Le premier PDO sur le nœud a une plus grande priorité que les deuxième, troisième et quatrième.
- Un PDO transmission est prioritaire sur un PDO réception dans le même nœud.
- Plus le numéro du nœud est bas, plus la priorité des PDO du nœud est grande.

Comment créer un PDO numéro 5

Le tableau ci-dessous décrit la procédure de création d'un 5e PDO pour un nœud donné.

Etape	Action
1	Configuration d'un PDO5 sur l'écran de configuration du nœud. **Résultat** : une fenêtre d'avertissement apparaît pour indiquer qu'il n'est pas possible de créer un nouveau PDO. C'est normal, étant donné ce qui est dit ci-dessus (un maximum de 4 PDO par nœud).
2	Sur l'écran de configuration du nœud, désactivez l'option Auto **COB-ID**. **Résultat** : vous pouvez maintenant écrire un nouveau COB-ID pour le PDO que vous souhaitez créer et donc définir les nouvelles priorités dans les messages transmis sur le bus CANopen. **Attention** : vous devez respecter l'unicité des COB-ID. Les numéros disponibles sont : ● TXPDO: 1664 à 1759 ● RXPDO: 1761 à 1792
3	Fermez la fenêtre de configuration du nœud et sauvegardez votre projet.

Glossaire



C

CAN

Controller Area Network : bus de terrain développé à l'origine pour l'automobile qui est maintenant utilisé dans de nombreux domaines, de l'industrie au tertiaire.

CiA

CAN in Automation : groupement international des utilisateurs et constructeurs de produits CAN.

COB

Communication OBject : unité de transport sur le bus CAN. Un COB est identifié par un identifiant unique codé sur 11 bits, [0, 2047]. Un COB contient au plus 8 octets de données. La priorité de transmission d'un COB est donnée par son identifiant, plus l'identifiant est faible plus le COB associé est prioritaire.

CRC

Cyclic Redundancy Checksum : checksum de redondance cyclique qui indique qu'aucun caractère n'a été "déformé" lors de la transmission de la trame.

CSMA/CA

Carrier Sense, Multiple Access / Collision Avoidance : méthode de gestion des communications sur un réseau caractérisant la couche liaison.

D

DIN

Deutsches Institut für Normung: institut de normalisation allemand.

DS

Draft Standard: document de spécification issu des travaux du groupement CIA.

E

EDS

Electronic Data Sheet: fichier de description de chaque équipement CAN (fourni par les constructeurs). Avec le logiciel de configuration Sycon, pour ajouter un équipement CAN sur le bus, il faut sélectioner l'EDS correspondant. Les EDS sont disponibles sur le site **http://www.can-cia.de** ou bien auprès du fournisseur du matériel.

L

Life Time

Life Time = Life Time factor x Guard Time .

LLC

Logical Link Control .

M

MAC

Medium Access Control .

MDI

Medium Dependent Interface .

MTBF

Mean Time Between Failure : temps moyen entre deux pannes.

O

OD

Object Dictionary : dictionnaire des objets reconnus par CAN. Un code en hexadécimal est donné à chaque type d'objet, le dictionnaire regroupe les codes de tous les objets.

P

PCMCIA

Personal Computer Memory Card International Association

PDO

Process Data Object : il existe les RPDO (Recieve PDO) et les TPDO (Transmit PDO).

PDU

Process Data Unit : il existe les APDU (Application PDU). Un PDU au niveau de la couche liaison est un APDU encapsulé par les entêtes et les octets caractérisant cette couche liaison.

PMA

Physical Medium Attachment.

S

SDO

Service Data Object : il existe les SSDO (Server SDO) et les CSDO (Client SDO).

T

TAP

Transmission Access Point : boitier de raccordement du bus.

Index

A

adressage
    topologique, *35*
affectation PDO, *29*

C

câbles de dérivation, *17*
COB-ID, *150*
codes d'erreur, *102*
compatibilité des processeurs, *30*
configuration, *36*
    étapes de configuration, *34*
contrôle d'erreur
    heartbeat, *129*
    surveillance des abonnés, *129*

D

diagnostic, *70*
diagnostics, *69*

L

longueurs de bus, *13*

M

mise au point, *67*
mise en route, *109*

N

NMT (gestion réseau), *29*
normes, *28*

P

PDO, *143*
programmation, *56*

R

READ_VAR, *57*
réglages des paramètres, *77*

S

SDO, *57*
SEND_REQ, *63*
structure de données de voie pour modules CANopen
    T_COM_CPP110, *91*
Structure des données de voie pour tous les modules
    IODDT, *87*
structure des données de voie pour tous les modules
    T_GEN_MOD, *98*

T

T_COM_CPP110, *91*

T_GEN_MOD, *98*
TSX CPP 110, *21*

V

vitesses de transmission, *13*

W

WRITE_VAR, *57*